frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 19 June 2019
doi: 10.3389/fams.2019.00030

Stochastic AUC Optimization Algorithms With Linear Convergence

Michael Natole Jr.[1], *Yiming Ying*[1]* and *Siwei Lyu*[2]

[1] Department of Mathematics and Statistics, University at Albany, State University of New York, Albany, NY, United States,
[2] Department of Computer Science, University at Albany, State University of New York, Albany, NY, United States

Area under the ROC curve (AUC) is a standard metric that is used to measure classification performance for imbalanced class data. Developing stochastic learning algorithms that maximize AUC over accuracy is of practical interest. However, AUC maximization presents a challenge since the learning objective function is defined over a pair of instances of opposite classes. Existing methods circumvent this issue but with high space and time complexity. From our previous work of redefining AUC optimization as a convex-concave saddle point problem, we propose a new stochastic batch learning algorithm for AUC maximization. The key difference from our previous work is that we assume that the underlying distribution of the data is uniform, and we develop a batch learning algorithm that is a stochastic primal-dual algorithm (SPDAM) that achieves a linear convergence rate. We establish the theoretical convergence of SPDAM with high probability and demonstrate its effectiveness on standard benchmark datasets.

Keywords: AUC maximization, imbalanced data, linear convergence, stochastic optimization, ROC curve

OPEN ACCESS

Edited by:
Carlos Mejía-Monasterio, Polytechnic University of Madrid, Spain

Reviewed by:
Junhong Lin, École Polytechnique Fédérale de Lausanne, Switzerland
Jinshan Zeng, Jiangxi Normal University, China

***Correspondence:**
Yiming Ying
yying@albany.edu

Specialty section:
This article was submitted to Mathematics of Computation and Data Science, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *08 March 2019*
Accepted: *27 May 2019*
Published: *19 June 2019*

Citation:
Natole M Jr, Ying Y and Lyu S (2019) Stochastic AUC Optimization Algorithms With Linear Convergence. Front. Appl. Math. Stat. 5:30. doi: 10.3389/fams.2019.00030

1. INTRODUCTION

Quantifying machine learning performance is an important issue to consider when designing learning algorithms. Many existing algorithms maximize accuracy, however, it can be a misleading performance metric for several reasons. First, accuracy assumes that an equal misclassification cost for positive and negative labeling. This assumption is not viable for many real world examples such as medical diagnosis and fraud detection [1]. Also, optimizing accuracy is not suitable for important learning tasks such as imbalanced classification. To overcome these issues, Area Under the ROC Curve (AUC) [2, 3] is a standard metric for quantifying machine learning performance. It is used in many real world applications, such as ranking and anomaly detection. AUC concerns the overall performance of a functional family of classifiers and quantifies their ability of correctly ranking any positive instance with regards to a randomly chosen negative instance. This combined with the fact that AUC is not effected by imbalanced class data makes AUC a more robust metric than accuracy [4]. We will discuss maximizing AUC in a batch learning setting.

Learning algorithms that maximize AUC performance have been developed in both batch and online settings. Previously, most algorithms optimizing AUC for classification [5–8] were for batch learning, where we assume all training data is available making those methods not applicable to streaming data. However, online learning algorithms [9–14], have been proven to be very efficient to deal with large-scale datasets and streaming data. The issue with these studies is that they focus on optimizing the misclassification error or its surrogate loss. These works all attempt to overcome the problem that AUC is based on the sum of pairwise losses between examples from different classes, making the objective function quadratic in the number of samples. Overcoming this issue is the challenge of designing algorithms to optimize the AUC score in either setting.

In this work, we present a new stochastic batch learning algorithm for AUC maximization, SPDAM. The algorithm is based on our previous work that we can reformulate AUC maximization as a stochastic saddle point problem with the inclusion of a regularization term [15]. However, the key difference from our previous work is that SPDAM assumes that the distribution is uniform and is solved as a stochastic primal dual algorithm [16]. The proposed algorithm results in a faster convergence rate than existing state-of-the-art algorithms. When evaluating on several standard bench mark datasets, SPDAM achieves performances that are on par with other state-of-the-art AUC optimization methods with a significant improvement in running time.

The paper is organized as follows: Section 2 discusses related work. Section 3 briefly reformulates AUC optimization as a saddle point problem. Section 4 exploits section 3 with the assumption that the distribution is a uniform distribution over the data and introduces SPDAM. Section 5 details the experiments. Finally, section 6 gives some final thoughts.

2. RELATED WORK

AUC has been studied extensively because it is an appropriate performance measure for when dealing with imbalanced data distributions for learning classification. Designing such algorithms that optimize AUC is a significant challenge because of the need for samples of opposite classes. An early work first maximized the AUC score directly by performing gradient descent constrained to a hypersphere [17]. Their algorithm used a differentiable approximation to the AUC score that was accurate and computationally efficient, being of the order of $\mathcal{O}(n)$, where n is the number of data observations. Another early work optimized the AUC score using support vector machines [6].

In more recent work [18–22], significant progress has been done to design online learning algorithms for AUC maximization. Online methods are desirable for evaluating streaming data since these methods update when new data is available. However, a limitation of these methods is that the previous samples used need to be stored. For iteration t and where the dimension of the data is d, this results in a space and time complexity of $\mathcal{O}(td)$. This is an undesirable property because these algorithms will not scale well for high-dimensional data as well as will require more resources. To overcome the quadratic nature of AUC, the problem of optimizing the AUC score can be reformulated as a sum of pairwise loss functions using hinge loss [19, 22]. The use of a buffer with size s was proposed. This lessens the complexity to $\mathcal{O}(sd)$. However, if the buffer size is not set sufficiently large this will impact the performance of the method.

Again, using the idea of reformulating AUC as a sum of pairwise loss functions was further expanded upon [18]. Using the square loss function instead of hinge loss, a key observation was made in which the mean and covariance statistics of the training data could be easily updated as new data becomes available. Unlike the previous work where s samples needed to be stored, these statistics only needed to be stored. However, this algorithm still results in scaling issues for high-dimensional data because storing the covariance matrix results in a quadratic complexity of $\mathcal{O}(d^2)$. The authors did make note of this issue and proposed using low-rank Gaussian matrices to approximate the covariance matrix. The approximation is not a general solution to the original problem and depends on whether the covariance matrix can be well approximated by low-rank matrices.

Work has been also been done to maximize AUC using batch methods. In Ding et al. [23], the authors propose an algorithm that uses an adaptive gradient method that uses the knowledge of historical gradients and that is less sensitive to parameter selection. The method proposed in Gultekin et al. [24] is based on a convex relaxation of the AUC function, but instead of using stochastic gradients, the algorithm uses the first and second order U-statistics of pairwise distances. A critical feature of this approach is that it is learning rate free as training the step size is a time consuming task.

More recently, work based on Ying et al. [25] has been expanded upon. The critical idea was the primal and dual variables introduced have distinct solutions. Two different works took advantage of this observation. The first work developed a primal dual style stochastic gradient method [26] while the other develops a stochastic proximal algorithm that can have non-smooth penalty functions [27, 28]. Both algorithms achieve a $\mathcal{O}(1/T)$ convergence rate up to a logarithmic term.

3. PROBLEM STATEMENT

First, consider $\mathcal{X} \subseteq \mathbb{R}^d$ to be the input space and $\mathcal{Y} = \{-1, +1\}$ the output space. For the training data, $\mathbf{z} = \{(x_i, y_i),\ i = 1, \ldots, n\}$, we assume to be *i.i.d.* and the samples are obtained from an unknown distribution ρ on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. As in Ying et al. [25], we restrict this work to the family of linear functions, i.e., $f(x) = \mathbf{w}^\top x$.

3.1. AUC Optimization

The ROC curve is the plot of the true positive rate vs. the false positive rate. The area under the ROC curve (AUC) for any scoring function $f : \mathcal{X} \to \mathbb{R}$ is equivalent to the probability of a positive sample ranking higher than a negative sample [3, 29]. It is defined as

$$\mathrm{AUC}(f) = \Pr(f(x) \geq f(x') | y = +1, y' = -1), \tag{1}$$

where (x, y) and (x', y') are independently drawn from ρ. The intent of AUC maximization is to find the optimal decision function f:

$$\begin{aligned} \arg\max_f \mathrm{AUC}(f) &= \arg\min_f \Pr(f(x) < f(x') | y = 1, y' = -1) \\ &= \arg\min_f \mathbb{E}\Big[\mathbb{I}_{[f(x') - f(x) > 0]} \big| y = 1, y' = -1\Big], \end{aligned} \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function. As in Ying et al. [25], define $p = \Pr(y = 1)$. Recall that the conditional expectation of a random variable $\xi(z)$ is defined by $\mathbb{E}[\xi(z) | y = 1] = \frac{1}{p} \iint \xi(z) \mathbb{I}_{y=1} d\rho(z)$. In (2), the indicator function is not continuous, and is usually replaced by a convex surrogate such

as the ℓ_2 loss $(1-(f(x)-f(x')))^2$ or the hinge loss $\big(1-(f(x)-f(x'))\big)_+$. We used the ℓ_2 loss for this work as it has been shown to be statistically consistent with AUC while the hinge loss is not [18, 30]. Letting λ be a regularization parameter, AUC maximization can be formulated by

$$\begin{aligned}&\operatorname*{argmin}_{\mathbf{w}} \mathbb{E}\Big[(1-\mathbf{w}^\top(x-x'))^2|y=1,y'=-1\Big]+\frac{\lambda}{2}\|\mathbf{w}\|^2,\\ &=\operatorname*{argmin}_{\mathbf{w}} \frac{1}{p(1-p)}\iint_{\mathcal{Z}\times\mathcal{Z}}(1-\mathbf{w}^\top(x-x'))^2\\ &\mathbb{I}_{[y=1,y'=-1]}d\rho(z)d\rho(z')+\tfrac{\lambda}{2}\|\mathbf{w}\|^2.\end{aligned} \tag{3}$$

where the samples (x,y) and (x',y') are independent. When ρ is a uniform distribution over training data $\mathbf{z}$, we obtain the empirical minimization (ERM) problem for AUC optimization studied in Gao et al. [18] and Zhao et al. [22]

$$\operatorname*{argmin}_{\mathbf{w}} \frac{1}{n^+n^-}\sum_{i=1}^{n}\sum_{j=1}^{n}(1-\mathbf{w}^\top(x_i-x_j))^2\mathbb{I}_{[y_i=1\wedge y_j=-1]}+\frac{\lambda}{2}\|\mathbf{w}\|^2, \tag{4}$$

where n^+ and n^- denote the numbers of instances in the positive and negative classes, respectively.

3.2. Equivalent Representation as a Saddle Point Problem (SPP)

As in Ying et al. [25], AUC optimization as in (3) can be represented as stochastic Saddle Point Problem (SPP) (e.g., [15]). A stochastic SPP is generally in the form of

$$\min_{u\in\Omega_1}\max_{\alpha\in\Omega_2}\Big\{f(u,\alpha):=\mathbb{E}[F(u,\alpha,\xi)]\Big\}, \tag{5}$$

where $\Omega_1\subseteq\mathbb{R}^d$ and $\Omega_2\subseteq\mathbb{R}^m$ are non-empty closed convex sets, ξ is a random vector with non-empty measurable set $\Xi\subseteq\mathbb{R}^p$, and $F:\Omega_1\times\Omega_2\times\Xi\to\mathbb{R}$. Here $\mathbb{E}[F(u,\alpha,\xi)]=\int_\Xi F(u,\alpha,\xi)d\Pr(\xi)$, and function $f(u,\alpha)$ is convex in $u\in\Omega_1$ and concave in $\alpha\in\Omega_2$. In general, u and α are referred to as the primal variable and the dual variable, respectively. In this work, we modified our formulation for AUC maximization to include a regularization term. We give a modified version of the result in Ying et al. [25] that includes the L^2 term. First, define $F:\mathbb{R}^d\times\mathbb{R}^3\times\mathcal{Z}\to\mathbb{R}$, for any $\mathbf{w}\in\mathbb{R}^d$, $a,b,\alpha\in\mathbb{R}$ and $z=(x,y)\in\mathcal{Z}$, by

$$\begin{aligned}F(\mathbf{w},a,b,\alpha;z)=&(1-p)(\mathbf{w}^\top x-a)^2\mathbb{I}_{[y=1]}+p(\mathbf{w}^\top x-b)^2\mathbb{I}_{[y=-1]}\\ &+2(1+\alpha)(p\mathbf{w}^\top x\mathbb{I}_{[y=-1]}-(1-p)\mathbf{w}^\top x\mathbb{I}_{[y=1]})\\ &-p(1-p)\alpha^2+\frac{\lambda}{2}\|\mathbf{w}\|^2.\end{aligned} \tag{6}$$

Equation (6) is similar as in our previous work [25]. The only difference is the inclusion of a regularization term. The main result still holds in a similar manner.

Theorem 3.1. *The AUC optimization (3) is equivalent to*

$$\min_{\substack{\mathbf{w}\in\mathbb{R}^d\\(a,b)\in\mathbb{R}^2}}\max_{\alpha\in\mathbb{R}}\Big\{f(\mathbf{w},a,b,\alpha):=\int_{\mathcal{Z}}F(\mathbf{w},a,b,\alpha;z)d\rho(z)\Big\}. \tag{7}$$

In addition, we can prove the following result.

Proposition 3.1. *For any saddle point $(\mathbf{w}^*,a^*,b^*,\alpha^*)$ of the SPP formulation (7), $\mathbf{w}^*$ is a minimizer of the original AUC optimization problem (3).*

Proof: Let $\bar{f}(\mathbf{w},a,b,\alpha)=1+\frac{\int_{\mathcal{Z}}F(\mathbf{w},a,b,\alpha;z)d\rho(z)}{p(1-p)}+\frac{\lambda}{2}\|\mathbf{w}\|^2$ and let $(\mathbf{w}^*,a^*,b^*,\alpha^*)$ be a saddle point of the problem

$$\min_{\substack{\mathbf{w}\in\mathbb{R}^d\\(a,b)\in\mathbb{R}^2}}\max_{\alpha\in\mathbb{R}}\bar{f}(\mathbf{w},a,b,\alpha).$$

Since the order of the two minimization [i.e., minimizing with respect to $\mathbf{w}$ and minimizing with respect to (a,b)] does not affect the result. This implies, for every fixed $\mathbf{w}$, (a^*,b^*,α^*) is a saddle point of the sub-problem

$$\min_{(a,b)\in\mathbb{R}^2}\max_{\alpha\in\mathbb{R}}\bar{f}(\mathbf{w},a,b,\alpha).$$

Notice from the proof for Theorem 3.1 that

$$\begin{aligned}&\mathbb{E}\Big[(1-\mathbf{w}^\top(x-x'))^2|y=1,y'=-1\Big]+\frac{\lambda}{2}\|\mathbf{w}\|^2=\\ &\min_{(a,b)\in\mathbb{R}^2}\max_{\alpha\in\mathbb{R}}\bar{f}(\mathbf{w},a,b,\alpha).\end{aligned} \tag{8}$$

Hence,

$$\mathbb{E}\Big[(1-\mathbf{w}^\top(x-x')^2|y=1,y'=-1\Big]+\frac{\lambda}{2}\|\mathbf{w}\|^2=\bar{f}(\mathbf{w},a^*,b^*,\alpha^*).$$

This further implies

$$\begin{aligned}&\min_{\mathbf{w}}\mathbb{E}\Big[(1-\mathbf{w}^\top(x-x'))^2|y=1,y'=-1\Big]+\frac{\lambda}{2}\|\mathbf{w}\|^2=\\ &\min_{\mathbf{w}}f(\mathbf{w},a^*,b^*,\alpha^*).\end{aligned} \tag{9}$$

As $\mathbf{w}^*$ is a minimizer of the righthand side of the Equation (9), $\mathbf{w}^*$ is also a minimizer of the lefthand side of the equation.

4. STOCHASTIC PRIMAL-DUAL ALGORITHM FOR AUC MAXIMIZATION

The algorithm developed in our previous work focused on the population objective of the saddle point problem (7). It is essentially an online projected gradient descent algorithm which has an optimal convergence rate $\mathcal{O}(1/\sqrt{t})$. This convergence rate is distribution-free, i.e., it holds true for any distribution ρ.

In this section, we are concerned with the case that the distribution ρ in (7) is a uniform distribution over the given data $\mathbf{z}=\{z_1,\ldots,z_n\}$. Denote by $\mathbb{N}_n=\{1,2,\ldots,n\}$ for any $n\in\mathbb{N}$. Now, when ρ is a uniform distribution over finite data $\{(x_i,y_i):i\in\mathbb{N}_n\}$, we can reformulate (4) as a SPP as in (5):

$$\min_{\substack{\mathbf{w}\in\mathbb{R}^d\\(a,b)\in\mathbb{R}^2}}\max_{\alpha\in\mathbb{R}}\frac{1}{n}\sum_{i\in\mathbb{N}_n}F(\mathbf{w},a,b,\alpha,z_i). \tag{10}$$

In this case, the AUC optimization is equivalent to the saddle point problem (10). For this special case, we will develop in this section a stochastic primal-dual algorithm for AUC optimization (10) which is able to converge with a linear convergence rate. To this end, we now consider the following general saddle point problem for AUC maximization

$$\min_{\mathbf{w},a,b}\max_{\alpha}\Big\{\frac{1}{n_+}\sum_{i\in\mathbb{N}_n}(\mathbf{w}^\top x_i-a)^2\mathbb{I}_{y_i=1}+\frac{1}{n_-}\sum_{i\in\mathbb{N}_n}(\mathbf{w}^\top x_i-b)^2\mathbb{I}_{y_i=-1} + 2(1+\alpha)\mathbf{w}^\top\Big[\frac{1}{n_-}\sum_{i\in\mathbb{N}_n}x_i\mathbb{I}_{y_i=-1}-\frac{1}{n_+}\sum_{i\in\mathbb{N}_n}x_i\mathbb{I}_{y_i=1}\Big]-\alpha^2 + \Omega(\mathbf{w})\Big\}, \tag{11}$$

where $\Omega(\mathbf{w})$ is a penalty term. If $\Omega(\mathbf{w}) = \mathbb{I}_{\|\mathbf{w}\|\le R}(\mathbf{w})$, the above formulation is equivalent to the saddle point formulation (10).

Before describing the detailed algorithm, we introduce some notations and slightly modify the saddle formulation (11). Specifically, denote by n_+ and n_- the numbers of samples in the positive and negative classes, respectively. In this discrete case $p = \frac{n_+}{n}$. Let $\mathbf{b} = \mathbf{m}_- - \mathbf{m}_+$ where $\mathbf{m}_+$ and $\mathbf{m}_-$ are the means of the positive and negative classes, respectively, i.e., $\mathbf{m}_+ = \frac{1}{n_+}\sum_{i\in\mathbb{N}_n}x_i\mathbb{I}_{y_i=1}$ and $\mathbf{m}_- = \frac{1}{n_-}\sum_{i\in\mathbb{N}_n}x_i\mathbb{I}_{y_i=-1}$. For any $i\in\mathbb{N}_n$, denote

$$\bar{x}_i=\frac{x_i-m_+}{\sqrt{2p}}\quad \text{if } y_i=1,\qquad \bar{x}_i=\frac{x_i-m_-}{\sqrt{2(1-p)}}\quad \text{if } y_i=-1. \tag{12}$$

Let $g(\mathbf{w}) = \frac{|\mathbf{b}^\top\mathbf{w}|^2}{2}+\mathbf{b}^\top\mathbf{w}+\Omega(\mathbf{w})$. To satisfy the hypothesis that g is a λ strong convex function, we will let $\Omega(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2$. Now we have the following reformulation of (11), based on which we will develop a stochastic primal-dual algorithm for AUC maximization.

Proposition 4.1. *Formulation (13) is equivalent to*

$$\min_{\mathbf{w}}\max_{\beta}\Big\{\frac{1}{n}\sum_{i\in\mathbb{N}_n}\beta_i\mathbf{w}^\top\bar{x}_i-\frac{\|\beta\|^2}{2}+g(\mathbf{w})\Big\}, \tag{13}$$

where $g:\mathbb{R}^d\to\mathbb{R}$ is defined, for any $\mathbf{w}\in\mathbb{R}^d$, by $g(\mathbf{w}) = \frac{|\mathbf{b}^\top\mathbf{w}|^2}{2}+\mathbf{b}^\top\mathbf{w}+\Omega(\mathbf{w})$.

Proof: By minimizing out a, b and α, formulation (11) is equivalent to

$$\min_{\mathbf{w}}\max_{\alpha}\Big\{\frac{1}{n_+}\sum_{i\in\mathbb{N}_n}(\mathbf{w}^\top(x_i-\mathbf{m}_+))^2\mathbb{I}_{y_i=1} + \frac{1}{n_-}\sum_{i\in\mathbb{N}_n}(\mathbf{w}^\top(x_i-\mathbf{m}_-))^2\mathbb{I}_{y_i=-1}+2\mathbf{b}^\top\mathbf{w}+|\mathbf{b}^\top\mathbf{w}|^2+\Omega(\mathbf{w})\Big\}.$$

Substituting (12) into the above equation yields the desired result.

Recall that $\kappa = \max\{\|x_i\|: i\in\mathbb{N}_n\}$. We can establish the following linear convergence rate of SPDAM.

TABLE 1 | Pseudo-code of Stochastic Primal-Dual Algorithm for AUC maximization.

Stochastic Primal-Dual Algorithm for AUC Maximization (SPDAM)
1. Choose parameters $\sigma>0$ and $\tau>0$
2. Initialize $\beta^{(0)}$ and $\mathbf{w}^{(0)}$. Let $\bar{\mathbf{w}}^{(0)}=\mathbf{w}^{(0)}$ and $u^{(0)}=\frac{1}{n}\sum_{i\in\mathbb{N}_n}\beta_i^{(0)}\bar{x}_i$.
3. **For** $t=0,\ldots,T-1$ **do**
Uniformly and randomly choose $I\subseteq\mathbb{N}_n$ of size m and execute the following updates:
$\beta_i^{(t+1)}=\begin{cases}\operatorname{argmax}_{\beta_i\in\mathbb{R}}\{\beta_i\langle\bar{\mathbf{w}}^{(t)},x_i\rangle-\frac{\lvert\beta_i\rvert^2}{2}-\frac{\lvert\beta_i-\beta_i^{(t)}\rvert^2}{2\sigma}\} & \text{if } i\in I\\ \beta_i^{(t)} & \text{otherwise.}\end{cases}$
$u^{(t+1)}=u^{(t)}+\frac{1}{n}\sum_{i\in I}(\beta_i^{(t+1)}-\beta_i^{(t)})x_i.$
$\bar{u}^{(t+1)}=u^{(t)}+\frac{n}{m}(u^{(t+1)}-u^{(t)}).$
$\mathbf{w}^{(t+1)}=\operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^d}\{\langle\bar{u}^{(t+1)},\mathbf{w}\rangle+g(\mathbf{w})+\frac{\lVert\mathbf{w}-\mathbf{w}^{(t)}\rVert^2}{2\tau}\}.$
$\bar{\mathbf{w}}^{(t+1)}=\mathbf{w}^{(t+1)}+\theta(\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}).$
4. **end for**
5. **Output:** $\mathbf{w}^{(T)}$ and $\beta^{(T)}$

Theorem 4.1. *Assume that g is λ-strongly convex. Let $(\mathbf{w}^*,\beta^*)$ be the saddle point of (13). If the parameter σ,τ and θ are chosen such that*

$$\sigma = \frac{(n-m)+\sqrt{(n-m)^2+4n\kappa^2m/\lambda}}{8m\kappa^2},\ \tau=\frac{1}{4\sigma\kappa^2}\ \textit{and}\quad \theta = 1-\frac{\lambda}{\lambda+2\sigma\kappa^2},$$

then, for any $t\ge 1$, the SPDAM algorithm achieves

$$\left(\frac{1}{m}+\frac{1}{4\sigma m}\right)\mathbb{E}\big[\|\beta^{(t+1)}-\beta^*\|^2\big]+\left(\lambda+\frac{1}{2\tau}\right)\mathbb{E}\big[\|\mathbf{w}^{(t+1)}-\mathbf{w}^*\|^2\big]+\frac{1}{4\tau}\mathbb{E}\big[\|\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\|^2\big] \le \theta^t\Big[\left(\frac{1}{m}+\frac{1}{2\sigma m}\right)\|\beta^{(0)}-\beta^*\|+\left(\lambda+\frac{1}{2\tau}\right)\|\mathbf{w}^{(0)}-\mathbf{w}^*\|^2\Big]. \tag{14}$$

Before we present the proof for the above theorem. It is useful to make some comments. Firstly, the proposed algorithm in **Table 1** is inspired by the stochastic primal-dual algorithm proposed in Yu et al.[16] and Zhang and Lin [31] which focused on Support Vector Machines (SVM) and logistic regression. Secondly, the algorithm SPDAM enjoys faster convergence over the stochastic projected gradient method in our previous work. However, the incremental primal-dual algorithm here, in contrast to the algorithm in **Table 1** which can deal with streaming data, is not an online learning algorithm, since it needs to know a priori the number of the samples, the ratio of the samples of positive class, and means of the positive and negative classes. We now will prove the main theorem. The following lemma is critical for proving Theorem 4.1.

Lemma 4.1. *For the updates in SPDAM, we have*

$$\left(\frac{1}{m}+\frac{1}{2\sigma m}\right)\mathbb{E}\Big[\|\beta^{(t+1)}-\beta^*\|^2\Big]=\left(\frac{1}{2\sigma m}+\frac{n-m}{nm}\right)\mathbb{E}\Big[\|\beta^{(t)}-\beta^*\|^2\Big]-\frac{1}{2\sigma m}\mathbb{E}\big[\|\beta^{(t+1)}-\beta^{(t)}\|^2\big]+\mathbb{E}\big[\langle\bar{u}^{(t+1)},\bar{\mathbf{w}}^{(t)}-\mathbf{w}^*\rangle\big], \tag{15}$$

and

$$\left(\lambda+\frac{1}{2\tau}\right)\mathbb{E}\big[\|\mathbf{w}^{(t+1)}-\mathbf{w}^*\|^2\big]\le\frac{1}{2\tau}\mathbb{E}\big[\|\mathbf{w}^{(t)}-\mathbf{w}^*\|^2\big]-\frac{1}{2\tau}\mathbb{E}\big[\|\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\|^2\big]-\mathbb{E}\big[\langle\bar{u}^{(t+1)},\mathbf{w}^{(t+1)}-\mathbf{w}^*\rangle\big]. \tag{16}$$

Proof: We first prove Equation (15). For any $i\in\mathbb{N}_n$, let $\tilde{\beta}_i$ be defined as

$$\tilde{\beta}_i=\operatorname*{argmax}_{\beta_i\in\mathbb{R}}\left\{\beta_i\langle\bar{\mathbf{w}}^{(t)},x_i\rangle-\frac{|\beta_i|^2}{2}-\frac{|\beta_i-\beta_i^{(t)}|^2}{2\sigma}\right\}.$$

Hence,

$$\frac{|\beta_i^{(t)}-\beta_i^*|^2}{2\sigma}+\frac{|\beta_i^*|^2}{2}-\beta_i^*\langle\bar{\mathbf{w}}^{(t)},x_i\rangle=\frac{|\beta_i^{(t)}-\tilde{\beta}_i|^2}{2\sigma}+\frac{|\tilde{\beta}_i|^2}{2}-\tilde{\beta}_i\langle\bar{\mathbf{w}}^{(t)},x_i\rangle+\left(\frac{1}{2}+\frac{1}{2\sigma}\right)|\tilde{\beta}_i-\beta_i^*|^2. \tag{17}$$

Observe, by the definition of the saddle point $(\mathbf{w}^*,\beta^*)$, that

$$\beta^*=\operatorname*{argmax}_{\beta_i}\left\{\beta_i\langle\mathbf{w}^*,x_i\rangle-\frac{|\beta_i|^2}{2}\right\}.$$

Consequently, $\tilde{\beta}_i\langle\mathbf{w}^*,x_i\rangle-\frac{|\tilde{\beta}_i|^2}{2}=\beta_i^*\langle\mathbf{w}^*,x_i\rangle-\frac{|\beta_i^*|^2}{2}-\frac{1}{2}|\tilde{\beta}_i-\beta_i^*|^2$ which implies that $\frac{|\tilde{\beta}_i|^2}{2}-\frac{|\beta_i^*|^2}{2}=(\tilde{\beta}_i-\beta_i^*)\langle\mathbf{w}^*,x_i\rangle+\frac{1}{2}|\tilde{\beta}_i-\beta_i^*|^2$. Putting this back into (17), we have

$$\frac{|\beta_i^{(t)}-\beta_i^*|^2}{2\sigma}+(\tilde{\beta}_i-\beta_i^*)\langle\bar{\mathbf{w}}^{(t)}-\mathbf{w}^*,x_i\rangle=\frac{|\beta_i^{(t)}-\tilde{\beta}_i|^2}{2\sigma}+\left(1+\frac{1}{2\sigma}\right)|\tilde{\beta}_i-\beta_i^*|^2. \tag{18}$$

Let $\mathcal{F}_t$ be the sigma field generated by all random variables defined before round t. Taking expectation conditioned over $\mathcal{F}_t$ implies that

$$\mathbb{E}\big(|\beta_i^{(t)}-\beta_i^{(t+1)}|^2|\mathcal{F}_t\big)=\frac{m}{n}|\tilde{\beta}_i-\beta_i^{(t)}|^2,$$
$$\mathbb{E}\big(|\beta_i^{(t+1)}-\beta_i^*|^2|\mathcal{F}_t\big)=\frac{m}{n}|\tilde{\beta}_i-\beta_i^*|^2+\frac{n-m}{n}|\beta_i^{(t)}-\beta_i^*|^2,$$
$$\mathbb{E}\big(|\beta_i^{(t+1)}|^2|\mathcal{F}_t\big)=\frac{m}{n}|\tilde{\beta}_i|^2+\frac{n-m}{n}|\beta_i^{(t)}|^2,$$
$$\mathbb{E}\big(\beta_i^{(t+1)}|\mathcal{F}_t\big)=\frac{m}{n}\tilde{\beta}_i+\frac{n-m}{n}\beta_i^{(t)}.$$

Using the above equalities to represent terms involving $\tilde{\beta}_i$ by $\beta_i^{(t+1)}$ on the righthand side of (18), we have

$$\left(\frac{1}{m}+\frac{1}{2\sigma m}\right)\mathbb{E}\big[|\beta_i^{(t+1)}-\beta_i^*|^2|\mathcal{F}_t\big]=\left(\frac{1}{2\sigma m}+\frac{n-m}{nm}\right)|\beta_i^{(t)}-\beta_i^*|^2-\frac{1}{2\sigma m}\mathbb{E}[\|\beta^{(t+1)}-\beta^{(t)}\|^2]+\mathbb{E}\Big[\langle\bar{\mathbf{w}}^{(t)}-\mathbf{w}^*,\frac{1}{m}(\beta_i^{t+1}-\beta_i^*)+\frac{1}{n}(\beta_i^{(t)}-\beta_i^*)x_i\rangle|\mathcal{F}_t\Big]$$

Taking the summation over $i\in\mathbb{N}_n$ and noticing that $\bar{u}^{(t+1)}=\frac{1}{m}\sum_{i\in\mathbb{N}_n}(\beta_i^{t+1}-\beta_i^*)x_i+\frac{1}{n}\sum_{i\in\mathbb{N}_n}(\beta^{(t)}-\beta_i^*)x_i$, we have

$$\left(\frac{1}{m}+\frac{1}{2\sigma m}\right)\mathbb{E}\big[|\beta^{(t+1)}-\beta^*|^2|\big]=\left(\frac{1}{2\sigma m}+\frac{n-m}{nm}\right)\mathbb{E}[\|\beta^{(t)}-\beta^*\|^2]-\frac{1}{2\sigma m}\mathbb{E}[\|\beta^{(t+1)}-\beta^{(t)}\|^2]+\mathbb{E}\big[\langle\bar{\mathbf{w}}^{(t)}-\mathbf{w}^*,\bar{u}^{(t+1)}\big],$$

which completes the proof of the first estimation (15).

Now we turn our attention to the proof of inequality (16). Indeed, by the definition of $\mathbf{w}^{(t+1)}$ and λ-strongly convexity of g, there holds

$$\langle\bar{u}^{(t+1)},\mathbf{w}^*\rangle+g(\mathbf{w}^*)+\frac{\|\mathbf{w}^{(t)}-\mathbf{w}^*\|^2}{2\tau}\ge\langle\bar{u}^{(t+1)},\mathbf{w}^{(t+1)}\rangle+g(\mathbf{w}^{(t+1)})+\frac{\|\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\|^2}{2\tau}+\left(\frac{\lambda}{2}+\frac{1}{2\tau}\right)\|\mathbf{w}^{(t+1)}-\mathbf{w}^*\|^2. \tag{19}$$

Let $u^*=\frac{1}{n}\sum_{i\in\mathbb{N}_n}\beta_i^*x_i$. By the definition of the saddle point $(\mathbf{w}^*,\beta^*)$, there holds

$$\langle u^*,\mathbf{w}^{(t+1)}\rangle+g(\mathbf{w}^{(t+1)})\ge\langle u^*,\mathbf{w}^*\rangle+g(\mathbf{w}^*)+\frac{\lambda}{2}\|\mathbf{w}^{(t+1)}-\mathbf{w}^*\|^2.$$

Adding the above inequality with (19) and arranging the terms yields that

$$\left(\lambda+\frac{1}{2\tau}\right)\|\mathbf{w}^{(t+1)}-\mathbf{w}^*\|^2\le\frac{\|\mathbf{w}^{(t)}-\mathbf{w}^*\|^2}{2\tau}-\frac{\|\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\|^2}{2\tau}-\langle\mathbf{w}^{(t+1)}-\mathbf{w}^*,\bar{u}^{(t+1)}-u^*\rangle.$$

This completes the proof of the lemma.

Now we are ready to prove Theorem 4.1 using Lemma 4.1.

Proof: Adding (15) and (16) together, we have

$$\begin{aligned}&\left(\frac{1}{m}+\frac{1}{2\sigma m}\right)\mathbb{E}\big[\|\beta^{(t+1)}-\beta^*\|^2\big]+\left(\lambda+\frac{1}{2\tau}\right)\\&\mathbb{E}\big[\|\mathbf{w}^{(t+1)}-\mathbf{w}^*\|^2\big]\\&\leq\left(\frac{1}{2\sigma m}+\frac{1}{m}-\frac{1}{n}\right)\mathbb{E}\big[\|\beta^{(t)}-\beta^*\|\big]\\&+\frac{1}{2\tau}\mathbb{E}\big[\|\mathbf{w}^{(t)}-\mathbf{w}^*\|^2\big]-\frac{1}{2\sigma m}\mathbb{E}\big[\|\beta^{(t+1)}-\beta^{(t)}\|^2\big]\\&-\frac{1}{2\tau}\mathbb{E}\big[\|\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\|^2\big]\\&+\mathbb{E}\big[\langle u^{(t)}-u^*+\frac{n}{m}(u^{(t+1)}-u^{(t)}),\bar{\mathbf{w}}^{(t)}-\mathbf{w}^{(t+1)}\rangle\big].\end{aligned}\tag{20}$$

By the definition of $u^{(t)}$, $u^{(t+1)}$ and $\bar{\mathbf{w}}^{(t)}$, we have

$$\begin{aligned}&\langle u^{(t)}-u^*+\frac{n}{m}(u^{(t+1)}-u^{(t)}),\bar{\mathbf{w}}^{(t)}-\mathbf{w}^{(t+1)}\rangle\\&=\theta\langle u^{(t)}-u^*,\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\rangle\\&-\langle u^{(t+1)}-u^*,\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\rangle\\&+\frac{n\theta}{m}\langle u^{(t+1)}-u^{(t)},\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\rangle\\&+\frac{n-m}{m}\langle u^{(t+1)}-u^{(t)},\mathbf{w}^{(t)}-\mathbf{w}^{(t+1)}\rangle.\end{aligned}$$

By the Cauchy-Schwartz inequality, letting $X=[x_1,x_2,\ldots,x_n]^\top$ and noticing that $\kappa^2\sigma=\frac{1}{4\tau}$ we have

$$\begin{aligned}n\langle u^{(t+1)}-u^{(t)},\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\rangle&=\left\langle\sum_{i\in K}(\beta_i^{(t+1)}-\beta_i^{(t)})x_i,\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\right\rangle\\&\leq\frac{\|\beta^{(t+1)}-\beta^{(t)}\|\kappa^2 m}{4\sigma\kappa^2 m}+\frac{\|\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\|^2 m}{4\tau}\\&=\frac{\|\beta^{(t+1)}-\beta^{(t)}\|}{4\sigma}+\frac{\|\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\|^2 m}{4\tau}.\end{aligned}\tag{21}$$

Likewise,

$$\begin{aligned}n\langle u^{(t+1)}&-u^{(t)},\mathbf{w}^{(t)}-\mathbf{w}^{(t+1)}\rangle\leq\frac{\|\beta^{(t+1)}-\beta^{(t)}\|}{4\sigma}\\&+\frac{\|\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\|^2 m}{4\tau}.\end{aligned}$$

Putting these estimations into (22) and arranging the terms yield that

$$\begin{aligned}&\left(\frac{1}{m}+\frac{1}{2\sigma m}\right)\mathbb{E}\big[\|\beta^{(t+1)}-\beta^*\|^2\big]+\left(\lambda+\frac{1}{2\tau}\right)\\&\mathbb{E}\big[\|\mathbf{w}^{(t+1)}-\mathbf{w}^*\|^2\big]+\frac{1}{2\tau}\mathbb{E}\big[\|\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\|^2\big]\\&+\mathbb{E}\big[\langle u^{t+1}-u^*,\mathbf{w}^{(t+1)}-\mathbf{w}^{(t)}\rangle\big]\\&\leq\left(\frac{1}{m}+\frac{1}{2\sigma m}-\frac{1}{n}\right)\mathbb{E}\big[\|\beta^{(t)}-\beta^*\|\big]+\frac{1}{2\tau}\mathbb{E}\big[\|\mathbf{w}^{(t)}-\mathbf{w}^*\|^2\big]\\&+\theta\left(\frac{1}{2\tau}\mathbb{E}\big[\|\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\|^2\big]+\mathbb{E}\big[\langle u^t-u^*,\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\rangle\big]\right).\end{aligned}\tag{22}$$

Choosing that $\sigma=\frac{(n-m)+\sqrt{(n-m)^2+4n\kappa^2 m/\lambda}}{8m\kappa^2}$, $\tau=\frac{1}{4\sigma\kappa^2}$ and $\theta=1-\frac{\lambda}{\lambda+2\sigma\kappa^2}$ implies that

$$\left(\frac{1}{m}+\frac{1}{2\sigma m}-\frac{1}{n}\right)=\theta\left(1+\frac{1}{2\sigma}\right)\ \text{and}\ \frac{1}{2\tau}=\theta\left(\lambda+\frac{1}{2\tau}\right).\tag{23}$$

Letting

$$\begin{aligned}\triangle_t=&\left(\frac{1}{m}+\frac{1}{2\sigma m}\right)\mathbb{E}\big[\|\beta^{(t)}-\beta^*\|^2\big]+\left(\lambda+\frac{1}{2\tau}\right)\mathbb{E}\big[\|\mathbf{w}^{(t)}-\mathbf{w}^*\|^2\big]\\&+\frac{1}{2\tau}\mathbb{E}\big[\|\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\|^2\big]+\mathbb{E}\big[\langle u^t-u^*,\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\rangle\big],\end{aligned}$$

we know from (22) and (23) that $\triangle_{t+1}\leq\theta\,\triangle_t$. Using the exactly argument as in (21), there holds

$$\begin{aligned}&|\langle u^t-u^*,\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\rangle|\leq\frac{\|\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\|^2}{4\tau}\\&+\frac{\|(\beta^{(t)}-\beta^*)^\top X\|^2}{n^2/\tau}\leq\frac{\|\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\|^2}{4\tau}+\frac{\|(\beta^{(t)}-\beta^*)^\top X\|}{4n\sigma\kappa^2}\\&\leq\frac{\|\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\|^2}{4\tau}+\frac{\|\beta^{(t)}-\beta^*\|}{4n\sigma},\end{aligned}\tag{24}$$

which implies, for any t, that

$$\begin{aligned}\triangle_t\geq&\left(\frac{1}{m}+\frac{1}{4\sigma m}\right)\mathbb{E}\big[\|\beta^{(t)}-\beta^*\|^2\big]\\&+\left(\lambda+\frac{1}{2\tau}\right)\mathbb{E}\big[\|\mathbf{w}^{(t)}-\mathbf{w}^*\|^2\big]+\frac{1}{4\tau}\mathbb{E}\big[\|\mathbf{w}^{(t)}-\mathbf{w}^{(t-1)}\|^2\big]\geq 0.\end{aligned}\tag{25}$$

Consequently,

$$\begin{aligned}\triangle_{t+1}\ &\leq\theta^t\triangle_0=\theta^t\left(\left(\frac{1}{m}+\frac{1}{2\sigma m}\right)\|\beta^{(0)}-\beta^*\|\right.\\&\left.+\left(\lambda+\frac{1}{2\tau}\right)\|\mathbf{w}^{(0)}-\mathbf{w}^*\|^2\right).\end{aligned}$$

Combining this with the inequality (25) yields the desired result.

5. EXPERIMENTS

In this section, we report the experimental evaluations of SPDAM and compare it with existing state-of-the-art learning algorithms for AUC optimization and convergence rate.

5.1. Comparison Algorithms

We conduct comprehensive studies by comparing the proposed algorithm with other AUC optimization algorithms for both online and batch scenarios. Specifically, the algorithms considered in our experiments include:

- **SPDAM**: The proposed stochastic primal-dual algorithm for AUC maximization.

- **regSOLAM**: The regularized online projected gradient descent algorithm for AUC maximization.
- **OPAUC**: The one-pass AUC optimization algorithm with square loss function [18].
- **OAMseq**: The OAM algorithm with reservoir sampling and sequential updating method [22].
- **OAMgra**: The OAM algorithm with reservoir sampling and online gradient updating method [22].
- **Online Uni-Exp**: Online learning algorithm which optimizes the (weighted) univariate exponential loss [7].
- **B-SVM-OR**: A batch learning algorithm which optimizes the pairwise hinge loss [32].
- **B-LS-SVM**: A batch learning algorithm which optimizes the pairwise square loss.

It should be noted that OAMseq, OAMgra, and OPAUC are the state-of-the-art methods for AUC maximization in online settings. The algorithm regSOLAM is a modified version of our previous work that includes a regularization term and it achieves a similar convergence with only modified constants. We also reformulate the bound R in terms of the regularization parameter λ. Assume $\kappa = \sup_{x\in\mathcal{X}} \|x\| < \infty$, and recall that $\|\mathbf{w}\| \leq R$. By assuming that $\mathbf{w}^*$ is the optimal $\mathbf{w}$ then we have the following:

$$\frac{\lambda}{2}\|\mathbf{w}^*\|^2 \leq \mathbb{E}\Big[(1-\mathbf{w}^\top(x-x'))^2|y=1, y'=-1\Big] + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

By letting $\mathbf{w} = \mathbf{0}$ and recalling that $\|\mathbf{w}\| \leq R$, we can very easily see that: $R \leq \sqrt{\frac{2}{\lambda}}$. We make these changes to ensure a fair comparison with SPDAM.

TABLE 2 | Basic information about the benchmark datasets used in the experiments.

Datasets	♯inst	♯feat	Datasets	♯inst	♯feat
diabetes	768	8	fourclass	862	2
german	1,000	24	splice	3,175	60
usps	9,298	256	a9a	32,561	123
mnist	60,000	780	acoustic	78,823	50
ijcnn1	141,691	22	covtype	581,012	54
sector	9,619	55,197	news20	15,935	62,061

5.2. Experimental Testbed and Setup

To examine the performance of the proposed SPDAM algorithm in comparison to state-of-the-art methods, we conduct experiments on 11 benchmark datasets. **Table 2** shows the details of each of the datasets. All of these datasets are available for download from the LIBSVM and UCI machine learning repository. Note that some of the datasets (*mnist*, *covtype*, etc.) are multi-class, which we converted to binary data by randomly partitioning the data into two groups, where each group includes the same number of classes.

For the experiments, the features were normalized by taking $x_i \leftarrow \frac{x_i - \text{mean}(x_i)}{\|x_i\|}$ for the large datasets and $x_i \leftarrow \frac{x_i}{\|x_i\|}$ for the small datasets (*diabetes*, *fourclass*, and *german*). For each dataset, the data is randomly partitioned into 5-folds (4 are for training and 1 is for testing). We generate this partition for each dataset 5 times. This results in 25 runs for each dataset for which we use to calculate the average AUC score and standard deviation. To determine the proper parameter for each dataset, we conduct 5-fold cross validation on the training sets to determine the parameter $\lambda \in 10^{[-5:1]}$ for SPDAM and for regSOLAM the learning rate $\zeta \in [1:9:100]$ and the regularization parameter $\lambda \in 10^{[-5:5]}$ were found by a grid search. The buffer size for OAMseq and OAMgra is 100 as suggested [22]. All experiments for SPDAM and regSOLAM were conducted with MATLAB.

5.3. Evaluation of SPDAM and regSOLAM on Benchmark Datasets

Classification performances on the testing dataset of all methods are given in **Table 3**. These results show that SPDAM and

TABLE 3 | Comparison of the testing AUC values (mean±std.) on the evaluated datasets.

Datasets	SPDAM	regSOLAM	OPAUC	OAM_{seq}	OAM_{gra}	online Uni-Exp	B-SVM-OR	B-LS-SVM
diabetes	0.8275 ± 0.0302	0.8140 ± 0.0330	0.8309 ± 0.0350	0.8264 ± 0.0367	0.8262 ± 0.0338	0.8215 ± 0.0309	0.8326 ± 0.0328	0.8325 ± 0.0329
fourclass	0.8223 ± 0.0275	0.8222 ± 0.0276	0.8310 ± 0.0251	0.8306 ± 0.0247	0.8295 ± 0.0251	0.8281 ± 0.0305	0.8305 ± 0.0311	0.8309 ± 0.0309
german	0.7959 ± 0.0265	0.7830 ± 0.0247	0.7978 ± 0.0347	0.7747 ± 0.0411	0.7723 ± 0.0358	0.7908 ± 0.0367	0.7935 ± 0.0348	0.7994 ± 0.0343
splice	0.9227 ± 0.0128	0.9237 ± 0.0090	0.9232 ± 0.0099	0.8594 ± 0.0194	0.8864 ± 0.0166	0.8931 ± 0.0213	0.9239 ± 0.0089	0.9245 ± 0.0092
usps	0.9854 ± 0.0019	0.9848 ± 0.0021	0.9620 ± 0.0040	0.9310 ± 0.0159	0.9348 ± 0.0122	0.9538 ± 0.0045	0.9630 ± 0.0047	0.9634 ± 0.0045
a9a	0.8967 ± 0.0032	0.8970 ± 0.0048	0.9002 ± 0.0047	0.8420 ± 0.0174	0.8571 ± 0.0173	0.9005 ± 0.0024	0.9009 ± 0.0036	0.8982 ± 0.0028
mnist	0.9552 ± 0.0011	0.9599 ± 0.0014	0.9242 ± 0.0021	0.8615 ± 0.0087	0.8643 ± 0.0112	0.7932 ± 0.0245	0.9340 ± 0.0020	0.9336 ± 0.0025
acoustic	0.8119 ± 0.0039	0.8114 ± 0.0035	0.8192 ± 0.0032	0.7113 ± 0.0590	0.7711 ± 0.0217	0.8171 ± 0.0034	0.8262 ± 0.0032	0.8210 ± 0.0033
ijcnn1	0.9132 ± 0.0016	0.9108 ± 0.0030	0.9269 ± 0.0021	0.9209 ± 0.0079	0.9100 ± 0.0092	0.9264 ± 0.0035	0.9337 ± 0.0024	0.9320 ± 0.0037
covtype	0.9409 ± 0.0011	0.9332 ± 0.0020	0.8244 ± 0.0014	0.7361 ± 0.0317	0.7403 ± 0.0289	0.8236 ± 0.0017	0.8248 ± 0.0013	0.8222 ± 0.0014
sector	0.9406 ± 0.0062	0.9734 ± 0.0036	0.9292 ± 0.0081	0.9163 ± 0.0087	0.9043 ± 0.0100	0.9215 ± 0.0034	–	–

To accelerate the experiments, the value for sector was determined after five runs instead of 25 for the other data sets. The performances of OPAUC, OAMseq, OAMgra, online Uni-Exp, B-SVM-OR, and B-LS-SVM were taken from Gao et al. [18].

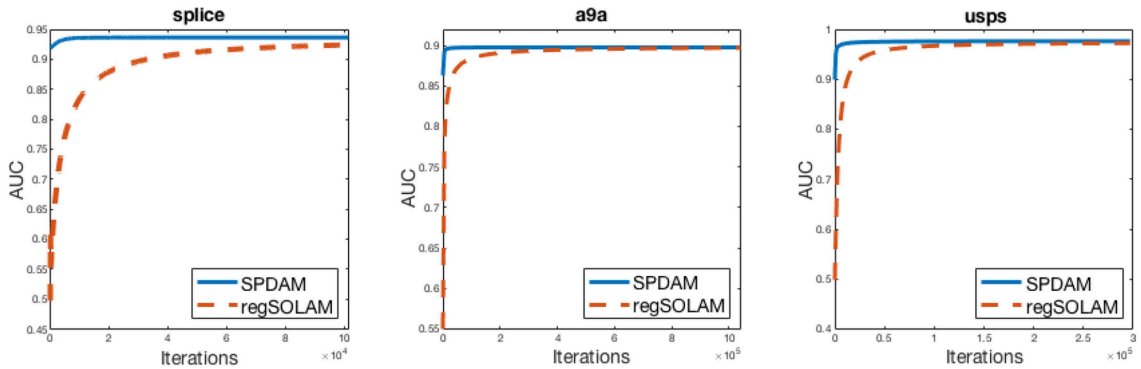

FIGURE 1 | AUC vs. Iteration curves of SPDAM against regSOLAM. For SPDAM, 10% of the data was chosen for a batch size. The optimal value of the parameter λ from SPDAM was used in regSOLAM.

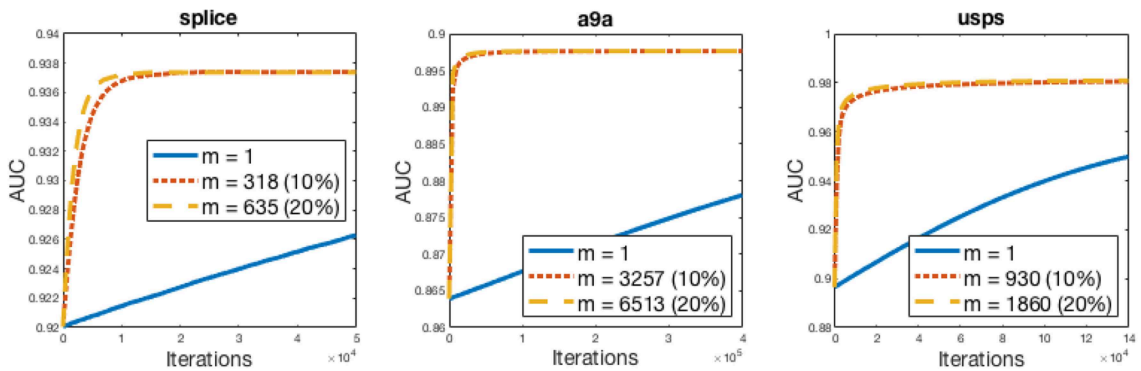

FIGURE 2 | AUC vs. Iteration curves of SPDAM algorithm for various batch sizes. The batch size is a percentage of the number of samples.

regSOLAM both achieve similar performances as other state-of-the-art online and offline methods based on AUC maximization. In some cases, SPDAM and regSOLAM perform better than some of the other online learning algorithms. There is a significant improvement in the text classification dataset *mnist* and *covtype*. The difference in performance of SPDAM and regSOLAM could be due to the fact since the data is randomly partitioned into two classes, the value of p could be resulting in a higher AUC score.

However, the main advantage of SPDAM over regSOLAM is the running time performance. SPDAM has a linear convergence rate while regSOLAM has a $\mathcal{O}(\frac{1}{\sqrt{t}})$ convergence. The theory tells us that SPDAM should be faster than regSOLAM. In **Figure 1**, we show AUC vs. Iterations for SPDAM against regSOLAM over 3 datasets. The figures show that SPDAM converges faster in comparison to regSOLAM, while maintaining a similar competitive performance as from **Table 3**.

In order to obtain this convergence rate, it is important to pick a large enough batch size (m). As from Theorem 4.1, the value of θ needs to be small for ensuring that SPDAM converges quickly. To ensure a fast convergence, the relationship between σ and θ should be examined. For θ to be small, σ should also be small which can be made possible by increasing the batch size m. If the batch size is too small, SPDAM will result in very poor performance. **Figure 2** demonstrates SPDAM on various batch sizes and shows that selecting a larger batch size ensures a faster rate of convergence. A batch size of 10% is sufficient so that SPDAM converges faster than regSOLAM.

6. CONCLUSION

In this paper, we proposed a stochastic primal-dual algorithm for AUC optimization [18, 22] based upon our previous work that AUC maximization is equivalent to a stochastic saddle point problem. By letting the distribution of ρ as in (7) be uniform, the proposed SPDAM algorithm is shown both theoretically and by experiments that the algorithm achieves a linear convergence rate. This makes SPDAM, given that a large enough batch size is used, faster than regSOLAM. If the batch size is not sufficiently large, SPDAM has poor performance.

There are several research directions for future work. First, the convergence was established using the duality gap associated with the stochastic SPP formulation (7). It would be interesting to establish the strong convergence of the output $\bar{\mathbf{w}}_T$ of the regSOLAM algorithm to its optimal solution of the actual AUC optimization problem (3). Secondly, the SPP formulation (3.1) holds for the least square loss. We do not know if the same formulation holds true for other loss functions such as the logistic regression or the hinge loss.

DATA AVAILABILITY

Publicly available datasets were analyzed in this study. This data can be found here: https://archive.ics.uci.edu/ml/index.php.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This work is supported by NSF grant (#1816227) and was supported by a grant from the Simons Foundation (#422504), and the Presidential Innovation Fund for Research and Scholarship (PIFRS) from SUNY Albany.

ACKNOWLEDGMENTS

This manuscript is a significant extension of work that first appeared at NIPS 2016 [25].

REFERENCES

1. Elkan C. The foundations of cost-sensitive learning. In: *International Joint Conference on Artificial Intelligence*. Vol. 17. Seattle, WA: Lawrence Erlbaum Associates Ltd (2001). p. 973–8.
2. Metz CE. Basic principles of ROC analysis. In: Freeman LM, Blaufox MD, editors. *Seminars in Nuclear Medicine*. Vol. 8. Amsterdam: Elsevier (1978). p. 283–98.
3. Hanley JA, McNeil BJ. The meaning and use of the area under of receiver operating characteristic (roc) curve. *Radiology*. (1982) 143:29–36. doi: 10.1148/radiology.143.1.7063747
4. Fawcett T. ROC graphs: notes and practical considerations for researchers. *Mach Learn*. (2004) 31:1–38.
5. Cortes C, Mohri M. AUC optimization vs. error rate minimization. In: *Neural Information Processing Systems*. Vancouver, BC (2003).
6. Joachims T. A support vector method for multivariate performance measures. In: *International Conference on Machine Learning*. Bonn (2005).
7. Kotlowski W, Dembczynski K, Hüllermeier E. Bipartite ranking through minimization of univariate loss. In: *International Conference on Machine Learning*. Bellevue, WA (2011).
8. Rakotomamonjy A. Optimizing area under Roc curve with SVMs. In: *1st International Workshop on ROC Analysis in Artificial Intelligence*. Valencia (2004).
9. Bach FR, Moulines E. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In: *Neural Information Processing Systems*. Granada (2011).
10. Bottou L, LeCun Y. Large scale online learning. In: *Neural Information Processing Systems*. (2003). Available online at: http://papers.nips.cc/paper/2365-large-scale-online-learning
11. Cesa-Bianchi N, Conconi A, Gentile C. On the generalization ability of on-line learning algorithms. *IEEE Trans Inform Theory*. (2004) 50:2050–7. doi: 10.1109/TIT.2004.833339
12. Rakhlin A, Shamir O, Sridharan K. Making gradient descent optimal for strongly convex stochastic optimization. In: *International Conference on Machine Learning*. Edinburgh (2012).
13. Ying Y, Pontil M. Online gradient descent learning algorithms. *Found Comput Math*. (2008) 8:561–96. doi: 10.1007/s10208-006-0237-y
14. Zinkevich M. Online convex programming and generalized infinitesimal gradient ascent. In: *International Conference on Machine Learning*. Washington, DC (2003).
15. Nemirovski A, Juditsky A, Lan G, Shapiro A. Robust stochastic approximation approach to stochastic programming. *SIAM J Optim*. (2009) 19:1574–609. doi: 10.1137/070704277
16. Yu W, Lin Q, Yang T. Doubly stochastic primal-dual coordinate method for regularized empirical risk minimization with factorized data. *CoRR*. (2015) abs/1508.03390. Available online at: http://arxiv.org/abs/1508.03390
17. Herschtal A, Raskutti B. Optimising area under the ROC curve using gradient descent. In: *Proceedings of the Twenty-First International Conference on Machine Learning*. Banff, AB: ACM (2004). p. 49.
18. Gao W, Jin R, Zhu S, Zhou ZH. One-pass AUC optimization. In: *International Conference on Machine Learning*. Atlanta, GA (2013).
19. Kar P, Sriperumbudur BK, Jain P, Karnick H. On the generalization ability of online learning algorithms for pairwise loss functions. In: *International Conference on Machine Learning*. Atlanta, GA (2013).
20. Wang Y, Khardon R, Pechyony D, Jones R. Generalization bounds for online learning algorithms with pairwise loss functions. In: *COLT*. Edinburgh (2012).
21. Ying Y, Zhou DX. Online pairwise learning algorithms. *Neural Comput*. (2016) 28:743–77. doi: 10.1162/NECO_a_00817
22. Zhao P, Hoi SCH, Jin R, Yang T. Online AUC maximization. In: *International Conference on Machine Learning*. Bellevue, WA (2011).
23. Ding Y, Zhao P, Hoi SCH, Ong Y. Adaptive subgradient methods for online AUC maximization. *CoRR*. (2016) abs/1602.00351. Available online at: http://arxiv.org/abs/1602.00351
24. Gultekin S, Saha A, Ratnaparkhi A, Paisley J. MBA: mini-batch AUC optimization. *CoRR*. (2018) abs/1805.11221. Available online at: http://arxiv.org/abs/1805.11221
25. Ying Y, Wen L, Lyu S. Stochastic online AUC maximization. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, editors. *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc. (2016). p. 451–9. Available online at: http://papers.nips.cc/paper/6065-stochastic-online-auc-maximization.pdf
26. Liu M, Zhang X, Chen Z, Wang X, Yang T. Fast stochastic AUC maximization with $O(1/n)$-convergence rate. In: Dy J, Krause A, editors. *Proceedings of the 35th International Conference on Machine Learning. vol. 80 of Proceedings of Machine Learning Research*. Stockholm: PMLR (2018). p. 3189–97. Available online at: http://proceedings.mlr.press/v80/liu18g.html
27. Rosasco L, Villa S, Vũ BC. Convergence of stochastic proximal gradient algorithm. *arXiv:14035074*. (2014).
28. Natole M Jr, Ying Y, Lyu S. Stochastic proximal algorithms for AUC maximization. In: Dy J, Krause A, editors. *Proceedings of the 35th International Conference on Machine Learning. vol. 80 of Proceedings of Machine Learning Research*. Stockholm: PMLR (2018). p. 3710–9. Available online at: http://proceedings.mlr.press/v80/natole18a.html
29. Clemencon S, Lugosi G, Vayatis N. Ranking and empirical minimization of U-statistics. *Ann Stat*. (2008) 36:844–74. doi: 10.1214/009052607000000910
30. Gao W, Zhou ZH. On the consistency of AUC pairwise optimization. In: *International Joint Conference on Artificial Intelligence*. Buenos Aires (2015).
31. Zhang Y, Lin X. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In: *International Conference on Machine Learning*. Lille (2015). p. 353–61.
32. Joachims T. Training linear SVMs in linear time. In: *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, PA (2006). p. 217–26.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.