Variance Reduced Stochastic Proximal Algorithm for AUC Maximization

Soham Dan(✉) and Dushyant Sahoo

University of Pennsylvania, Philadelphia, USA
{sohamdan,sadu}@seas.upenn.edu

Abstract. Stochastic Gradient Descent has been widely studied with classification accuracy as a performance measure. However, these stochastic algorithms are not applicable when non-decomposable pairwise performance measures are used, such as Area under the ROC curve (AUC), a standard performance metric used when the classes are imbalanced. Several algorithms have been proposed for optimizing AUC as a performance metric, one of the recent being a Stochastic Proximal Gradient Algorithm (SPAM). However, the downside of stochastic gradient descent is that it suffers from high variance leading to very slow convergence. Several variance reduced methods have been proposed with faster convergence guarantees than vanilla stochastic gradient descent to combat this issue. Again, these variance reduced methods are not applicable when non-decomposable performance measures are used. In this paper, we develop a Variance Reduced Stochastic Proximal algorithm for AUC Maximization (VRSPAM) that combines the two areas of analyzing non-decomposable performance metrics with and optimization efforts to guarantee faster convergence. We perform an in-depth theoretical and empirical analysis to demonstrate that our algorithm converges faster than existing state-of-the-art algorithms for the AUC maximization problem.

Keywords: Optimization · AUC · Variance reduction

1 Introduction

With the wide application of machine learning, there has been significant focus in recent times on applications that involve class imbalance—the case where one of the classes (the majority class) occurs much more frequently than the other class (the minority class) [6]. A concrete example is a medical diagnosis for a rare disease where far fewer instances from the disease class are observed than the healthy class. Traditional classification accuracy is not an appropriate performance metric in this setting, as predicting the majority class will give a high classification accuracy, even if the model always gives the wrong prediction on the minority class. To overcome this drawback, the Area under the ROC

S. Dan and D. Sahoo—Equal Contribution

N. Oliver et al. (Eds.): ECML PKDD 2021, LNAI 12977, pp. 184–199, 2021.
https://doi.org/10.1007/978-3-030-86523-8_12

curve (AUC) [7] is used as a standard metric for quantifying the performance of a binary classifier in this setting. AUC measures the ability of a family of classifiers to correctly rank an example from the positive class with respect to a randomly selected example from the negative class.

Several algorithms have been proposed for AUC maximization in the batch setting, where all the training data is assumed to be available at the beginning [12,25]. However, this assumption is unrealistic in several cases, especially for streaming data analysis, where examples are observed one at a time. For the usual classification accuracy metric, there exists *online algorithms* for such a streaming setting where the per iteration complexity is low [18,21]. However, despite several studies on online algorithms for classification accuracy, the case of maximizing AUC as a performance measure has been looked at only recently [14,26]. The main challenge for optimizing the AUC metric in the online setting is the pairwise nature of the AUC metric which, compared to classification accuracy, does not decompose over individual instances. In the AUC maximization framework, in each step the algorithm needs to pair the current datapoint with all previously observed datapoints leading to $\mathcal{O}(td)$ space and time complexity at step t, where the dimension of the instance space is d. The problem was not alleviated by the technique of buffering [14,26] since, good generalization performance depends on maintaining a large buffer.

From an optimization perspective, the AUC metric is non-convex and thus hard to optimize. Instead, it is attractive to optimize the convex surrogate, which is consistent, such as the pairwise squared surrogate [1,10,16]. Recently, [24] reformulated the pairwise squared loss surrogate of AUC as a saddle point problem and gave an algorithm that has a convergence rate of $\mathcal{O}(\frac{1}{\sqrt{t}})$. However, they only consider smooth regularization (penalty) terms such as Frobenius norm. Further, their convergence rate is sub-optimal to what stochastic gradient descent (SGD) achieves with classification accuracy as a performance measure $\mathcal{O}(\frac{1}{t})$. [17] improves on this with a stochastic proximal algorithm for AUC maximization, which under assumptions of strong convexity can achieve a convergence rate of $\mathcal{O}(\frac{\log t}{t})$ and has per iteration complexity of $\mathcal{O}(d)$ i.e., one datapoint and applies to general, non-smooth regularization terms.

Although [17] improves convergence for surrogate-AUC maximization, it still suffers from a high variance of the gradient in each iteration. Due to the large variance in random sampling, the stochastic gradient algorithm wastes time bouncing around, leading to worse performance and a slower sub-linear convergence rate of $\mathcal{O}(\frac{1}{t})$ (even if we ignore the $\log(t)$ term). Thus, we have the following trade-off: low per iteration complexity for the stochastic algorithm but slow convergence contrasted with high per iteration complexity and fast convergence for full gradient descent. Thus, it will take longer to get a good approximation of the solution to the AUC optimization problem if we employ the algorithm proposed by [17]. It is precisely this problem that we tackle in this paper: can we design an algorithm for AUC optimization that also enjoys fast convergence (potentially by controlling the variance of the iterates from the stochastic gradient algorithm).

In the relatively well-studied context of classification accuracy, techniques to reduce the variance of SGD have been proposed—SAG [19], SDCA [20], SVRG [13]. While SAG and SDCA require the storage of all the gradients and dual variables respectively, for complex models SVRG enjoys the same fast convergence rates as SDCA and SAG but has a much simpler analysis and does not require storage of gradients. This allows SVRG to be applicable in complex problems where the storage of all gradients would be infeasible.

Several works have explored ways to apply SVRG on classification problems involving a regularizer: the overall objective consists of the sum of a regularizer term and the average of several smooth component function terms in SVRG. Two simple strategies commonly used are the Proximal Full Gradient and the Proximal Stochastic Gradient method. While the Proximal Stochastic Gradient is much faster since it computes only the gradient of a single component function per iteration, it convergences much slower than the Proximal Full Gradient method, alluding to the same trade-off we mentioned earlier. The proximal gradient methods can be viewed as a particular case of splitting methods [2,3]. However, both the proximal methods do not fully exploit the problem structure. Proximal SVRG [22] is an extension of the SVRG [13] technique and can be used whenever the objective function is composed of two terms- the first term is an average of smooth functions (decomposable across the individual instances), and the second term admits a simple proximal mapping. Prox-SVRG needs far fewer iterations to achieve the same approximation ratio than the proximal full and stochastic gradient descent methods. However, all the existing techniques discussed that guarantee faster convergence by controlling the variance, including Prox-SVRG and the proximal full and stochastic gradient descent methods, all have a very restrictive assumption - they require the metric and the loss function to be decomposable over instances (for example, classification accuracy and the corresponding decomposable pointwise surrogate loss functions) and are not directly applicable to non-decomposable pairwise loss functions as in surrogate-AUC optimization (refer to Sect. 2); this is the gap that we close in this paper.

In this paper, we present Variance Reduced Stochastic Proximal algorithm for AUC Maximization (VRSPAM). VRSPAM builds upon previous work for surrogate-AUC maximization by using the SVRG algorithm. We provide theoretical analysis for the VRSPAM algorithm showing that it achieves a linear convergence rate with a fixed step size (much faster than SPAM [17], which has a sub-linear convergence rate and a decreasing step size). Also, the theoretical analysis provided in this paper simplifies the convergence analysis of SPAM. We perform numerical experiments to show that the VRSPAM algorithm converges significantly faster than SPAM.

2 AUC Formulation

The AUC score associated with a linear scoring function $g(x) = \mathbf{w}^T x$, is defined as the probability that the score of a randomly chosen positive example is higher than a randomly chosen negative example [5,11] and is denoted by AUC($\mathbf{w}$). If $z = (x, y)$ and $z' = (x', y')$ are drawn independently from an unknown distribution $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, then

$$\begin{aligned} \text{AUC}(\mathbf{w}) &= Pr(\mathbf{w}^T x \geq \mathbf{w}^T x' | y = 1, y' = -1) \\ &= \mathbb{E}[\mathbb{I}_{\mathbf{w}^T(x-x') \geq 0} | y = 1, y' = -1] \end{aligned}$$

Since AUC($\mathbf{w}$) in the above form is not convex because of the 0–1 loss, it is a common practice to replace this by a convex surrogate loss. In this paper, we focus on the least square loss which is known to be consistent (consistency of a surrogate loss function w.r.t the AUC metric means that, maximizing the surrogate function also maximizes the AUC).
Let $f(\mathbf{w}) = p(1-p)\mathbb{E}[(1 - \mathbf{w}^T(x - x'))^2 | y = 1, y' = -1]$ and Ω be the convex regularizer where $p = Pr(y = +1)$ and $1 - p = Pr(y = -1)$ are the class priors. We consider the following objective for surrogate-AUC maximization :

$$\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w}) + \Omega(\mathbf{w}) \tag{1}$$

The form for $f(\mathbf{w})$ follows from the definition of AUC : expected pairwise loss between a positive instance and a negative instance. Throughout this paper we assume

1. Ω is β strongly convex i.e. for any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$,

$$\Omega(\mathbf{w}) \geq \Omega(\mathbf{w}') + \partial\Omega(\mathbf{w}')^T(\mathbf{w} - \mathbf{w}') + \frac{\beta}{2}\|\mathbf{w} - \mathbf{w}'\|^2$$

2. $\exists M$ such that $\|x\| \leq M \ \forall x \in \mathcal{X}$.

In this paper we have used Frobenius norm $\Omega(\mathbf{w}) = \beta\|\mathbf{w}\|^2$ and Elastic Net $\Omega(\mathbf{w}) = \beta\|\mathbf{w}\|^2 + \nu\|\mathbf{w}\|_1$ as the convex regularizers where $\beta, \nu \neq 0$ are the regularization parameters.

It is important to note that standard stochastic gradient based algorithms cannot be applied to Eq. 1 directly, because of the pairwise nature of $f(\cdot)$. Instead we will use a reformulation that will allows us to apply stochastic gradient descent to find the optimum value of w. We write Eq. 1 in a pointwise manner rather than the above pairwise form, as originally proposed in [17], as follows:

$$\min_{\mathbf{w},a,b} \max_{\zeta \in \mathbb{R}} \mathbb{E}[F(\mathbf{w}, a, b, \zeta; z)] + \Omega(\mathbf{w}) \tag{2}$$

where the expectation is with respect to $z = (x, y)$ and

$$\begin{aligned} F(\mathbf{w},a,b,\zeta;z) = (1-p)(\mathbf{w}^T x - a)^2 \mathbb{I}_{[y=1]} + p(\mathbf{w}^T x - b)^2 \mathbb{I}_{[y=-1]} + \\ 2(1+\zeta)\mathbf{w}^T x(p\mathbb{I}_{[y=-1]} - (1-p)\mathbb{I}_{[y=1]}) - p(1-p)\zeta^2 \end{aligned}$$

Thus, $f(\mathbf{w}) = \min_{a,b} \max_{\zeta \in R} \mathbb{E}[F(\mathbf{w}, a, b, \zeta; z)]$. The optimal choices for a, b, ζ satisfy :

$$a(\mathbf{w}) = \mathbf{w}^T \mathbb{E}[x|y = 1]$$
$$b(\mathbf{w}) = \mathbf{w}^T \mathbb{E}[x|y = -1]$$
$$\zeta(\mathbf{w}) = \mathbf{w}^T (\mathbb{E}[x'|y' = -1] - \mathbb{E}[x|y = 1])$$

It is important to note here that we differentiate the objective function only with respect to $\mathbf{w}$ and do not compute the gradient with respect to the other parameters (a, b, ζ) which themselves depend on $\mathbf{w}$. Since, a, b, ζ are expressible in a closed-form, stochastic gradient algorithms can now be applied to Eq. 2. This is the SPAM algorithm [17].

3 Method

In the previous section, we discussed a stochastic gradient based algorithm for AUC maximization, that uses an alternative formulation of the objective, to make it decomposable. However, the SPAM algorithm suffers from very slow convergence in most real world problems which are high dimensional and consists of a large number of instances. The major issue that slows down convergence for SGD is the decay of the step size to 0 as the iteration increase. This is a necessary evil for mitigating the effect of variance introduced by random sampling in SGD. Thus, in this paper we directly attack the variance problem for SGD in the AUC maximization framework. We apply the Prox-SVRG method on the reformulation of AUC to derive the proximal SVRG algorithm for AUC maximization described in Algorithm 1. We store a $\tilde{\mathbf{w}}$ after every m Prox-SGD iterations that is progressively closer to the optimal $\mathbf{w}$ (essentially an estimate of the optimal value of (1). Full gradient $\tilde{\boldsymbol{\mu}}$ is computed whenever $\tilde{\mathbf{w}}$ gets updated i.e. after every m iterations of Prox-SGD:

$$\tilde{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^{n} G(\tilde{\mathbf{w}}, z_i)$$

where $G(\mathbf{w}; z) = \partial_{\mathbf{w}} F(\mathbf{w}, a(\mathbf{w}), b(\mathbf{w}), \zeta(\mathbf{w}); z)$, n is the number of samples and $\tilde{\boldsymbol{\mu}}$ is used to update next m gradients.
Next m iterations are initialized by $\mathbf{w}_0 = \tilde{\mathbf{w}}$. For each iteration, we randomly pick $i_t \in \{1, ..., n\}$ and compute

$$\hat{\mathbf{w}}_t = \mathbf{w}_{t-1} - \eta \mathbf{v}_t$$

where $\mathbf{v}_t = G(\mathbf{w}_{t-1}, z_{i_t}) - G(\tilde{\mathbf{w}}, z_{i_t}) + \tilde{\boldsymbol{\mu}}$ and then the proximal step is taken

$$\mathbf{w}_t = \text{prox}_{\eta,\Omega}(\hat{\mathbf{w}}_t)$$

Notice that if we take expectation of $G(\tilde{\mathbf{w}}, z_{i_t})$ with respect to i_t we get $\mathbb{E}[G(\tilde{\mathbf{w}}, z_{i_t})] = \tilde{\boldsymbol{\mu}}$. Now if we take expectation of $\mathbf{v}_t$ with respect to i_t conditioned on $\mathbf{w}_{t-1}$, we can get the following:

$$\begin{aligned}\mathbb{E}[\mathbf{v}_t|\mathbf{w}_{t-1}] &= \mathbb{E}[G(\mathbf{w}_{t-1}, z_{i_{t-1}})] - \mathbb{E}[G(\tilde{\mathbf{w}}, z_{i_{t-1}})] + \tilde{\boldsymbol{\mu}} \\ &= \frac{1}{n}\sum_{i=1}^{n} G(\tilde{\mathbf{w}}_{t-1}, z_i)\end{aligned}$$

Hence the modified direction $\mathbf{v}_t$ is the stochastic gradient of G at $\mathbf{w}_{t-1}$. However, the variance $\mathbb{E}\|\mathbf{v}_t - \partial f(\mathbf{w}_{t-1})\|^2$ can be much smaller than $\mathbb{E}\|G(\mathbf{w}_{t-1}, z_{i_t}) - \partial f(\mathbf{w}_{t-1})\|^2$, shown in Sect. 4.1. We will also show that the variance goes to 0 as the algorithm converges. Thus, this is a multi-stage scheme to explicitly reduce the variance of the modified proximal gradient.

Algorithm 1. Proximal SVRG for AUC maximization

INPUT Constant step size η and update frequency m
INITIALIZE $\tilde{\mathrm{W}}_0$
for $s = 1, 2, ...$ **do**
  $\tilde{\mathbf{w}} = \tilde{\mathbf{w}}_{s-1}$
  $\tilde{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n} G(\tilde{\mathbf{w}}, z_i)$
  $\mathbf{w}_0 = \tilde{\mathbf{w}}$
  for $t = 1, 2, ..., m$ **do**
    Randomly pick $i_t \in \{1, .., n\}$ and update weight
    $\hat{\mathbf{w}}_t = \mathbf{w}_{t-1} - \eta(G(\mathbf{w}_{t-1}, z_{i_t}) - G(\tilde{\mathbf{w}}, z_{i_t}) + \tilde{\boldsymbol{\mu}})$
    $\mathbf{w}_t = \mathrm{prox}_{\eta\Omega}(\hat{\mathbf{w}}_t)$
  $\tilde{\mathbf{w}}_s = \mathbf{w}_m$

4 Convergence Analysis

In this section, we formally analyze the convergence rate of VRSPAM. We first present some lemmas which will be used for proving the Theorem 1 which is the main theorem proving the geometric convergence of Algorithm 1. Lemma 1 states that
$\partial_{\mathbf{w}} F(\mathbf{w}_t, a(\mathbf{w}_t), b(\mathbf{w}_t), \alpha(\mathbf{w}_t); z_t)$ is an unbiased estimator of the true gradient. As we are not calculating the true gradient in VRSPAM, we need the following lemma to prove the convergence result.

Lemma 1 [17]**.** Let $\mathbf{w}_t$ be given by VRSPAM in Algorithm 1. Then, we have

$$\partial f(\mathbf{w}_t) = \mathbb{E}_{z_t}[\partial_{\mathbf{w}} F(\mathbf{w}_t, a(\mathbf{w}_t), b(\mathbf{w}_t), \alpha(\mathbf{w}_t); z_t)]$$

This lemma is directly applicable in VRSPAM since the proof of the lemma hinges on the objective function reformulation and not on the algorithm specifics.

The next lemma provides an upper bound on the norm of the difference of gradients at different time steps.

Lemma 2 [17]**.** Let $\mathbf{w}_t$ be described as above. Then, we have

$$\|G(\mathbf{w}_{t'}, z_t) - G(\mathbf{w}_{t''}, z_t)\| \leq 8M^2\|\mathbf{w}_{t'} - \mathbf{w}_{t''}\|$$

Proof.

$$\begin{aligned}\|G(\mathbf{w}_{t'}; z_t) - G(\mathbf{w}_{t''}; z_t)\| &\leq 4M^2p\|\mathbf{w}_{t'} - \mathbf{w}_{t''}\|\mathbb{1}_{[y_t=-1]} \\ &+ 4M^2p\|\mathbf{w}_{t'} - \mathbf{w}_{t''}\|\mathbb{1}_{[y_t=-1]} + 4M^2(1-p)\|\mathbf{w}_{t'} - \mathbf{w}_{t''}\|\mathbb{1}_{[y_t=1]} \\ &+ 4M^2|p - \mathbb{1}_{[y_t=1]}|\|\mathbf{w}_{t'} - \mathbf{w}_{t''}\| \\ &\leq 8M^2\|\mathbf{w}_{t'} - \mathbf{w}_{t''}\|\end{aligned}$$

The proof directly follows by writing out the difference and using the second assumption on the boundedness of $\|x\|$.

We now present and prove a key result that will be necessary in showing convergence in Theorem 1

Lemma 3. *Let* $C = \frac{1+128M^4\eta^2}{(1+\eta\beta)^2}$ *and* $D = \frac{128M^4\eta^2}{(1+\eta\beta)^2}$*; if* $\eta \leq \frac{\beta}{128M^4}$ *then* $C^m + DC\frac{C^m-1}{C-1} \leq 1$ *holds true.*

Proof. We start with:

$$\begin{aligned}&\eta \leq \frac{\beta}{128M^4} \\ \Rightarrow &128M^4\eta^2 \leq \eta\beta \\ \Rightarrow &128M^4\eta^2(2 + 128M^4\eta^2) \leq \eta\beta(2 + 1\eta\beta) \\ \Rightarrow &128M^4\eta^2 + (128M^4\eta^2)^2 \leq (\eta\beta)^2 + 2\eta\beta - 128M^4\eta^2 \\ \Rightarrow &128M^4\eta^2 \leq \frac{(1+\eta\beta)^2 - 1 - 128M^4\eta^2}{1 + 128M^4\eta^2} \\ \Rightarrow &128M^4\eta^2 \leq \frac{1 - \frac{1+128M^4\eta^2}{(1+\eta\beta)^2}}{\frac{1+128M^4\eta^2}{(1+\eta\beta)^2}}\end{aligned}$$

Substituting values of C and D and using the condition that $D \leq 128M^4\eta^2$, we get

$$\begin{aligned}&\Rightarrow D \leq \frac{1-C}{C} \\ &\Rightarrow DC\frac{C^m-1}{C-1} \leq 1 - C^m \\ &\Rightarrow C^m + DC\frac{C^m-1}{C-1} \leq 1\end{aligned}$$

Now we present and prove the main theorem of this paper that gives the convergence rate of Algorithm 1 and its analysis.

Theorem 1. *Consider* VRSPAM *(Algorithm 1) and let* $\mathbf{w}^* = arg\,min_{\mathbf{w}} f(\mathbf{w}) + \Omega(\mathbf{w})$*; if* $\eta < \frac{\beta}{128M^4}$*, then the following inequality holds true*

$$\alpha = C^m + DC\frac{C^m - 1}{C - 1} < 1$$

and we have the geometric convergence in expectation:

$$\mathbb{E}[\|\tilde{\mathbf{w}}_s - \mathbf{w}^*\|^2] \leq \alpha^s \mathbb{E}[\|\mathbf{w_0} - \mathbf{w}^*\|^2]$$

For proving the above theorem, first we upper bound the variance of the gradient step and show that it approaches zero as $\mathbf{w_s}$ approaches $\mathbf{w}^*$.

4.1 Bounding the Variance

In this section, we will derive a bound on the variance of the modified gradient $\mathbf{v}_t = G(\mathbf{w}_{t-1}, z_{i_t}) - G(\tilde{\mathbf{w}}, z_{i_t}) + \tilde{\boldsymbol{\mu}}$. We first present a lemma that will help derive the bound in Lemma 5.

Lemma 4. *Consider* VRSPAM *(Algorithm 1), then* $\mathbb{E}[\|\mathbf{v}_t - \partial f(\mathbf{w}^*)]\|^2$ *is upper bounded as:*

$$\mathbb{E}[\|\mathbf{v}_t - \partial f(\mathbf{w}^*)]\|^2] \leq 2(8M^2)^2\|\mathbf{w}_{t-1} - \mathbf{w}^*\|^2 + 2(8M^2)^2\|\tilde{\mathbf{w}} - \mathbf{w}^*\|^2$$

Proof. Let the variance reduced update be denoted as $\mathbf{v}_t = G(\mathbf{w}_{t-1}, z_{i_t}) - G(\tilde{\mathbf{w}}, z_{i_t}) + \tilde{\boldsymbol{\mu}}$.
As we know $\mathbb{E}[\mathbf{v}_t] = \partial f(\mathbf{w}_{t-1})$, the variance of $\mathbf{v}_t$ can be written as below

$$\begin{aligned}\mathbb{E}[\|\mathbf{v}_t - \partial f(\mathbf{w}^*))\|^2] \leq & 2\mathbb{E}[\|G(\mathbf{w}^*, z_{i_t}) - G(\tilde{\mathbf{w}}, z_{i_t}) + \tilde{\boldsymbol{\mu}} - \partial f(\mathbf{w}^*))\|^2] \\ & + 2\mathbb{E}[\|G(\mathbf{w}_{t-1}, z_{i_t}) - G(\mathbf{w}^*, z_{i_t})\|^2]\end{aligned}$$

Also, $\mathbb{E}[G(\mathbf{w}^*, z_{i_t}) - G(\tilde{\mathbf{w}}, z_{i_t})] = \partial f(\mathbf{w}^*) - \partial f(\tilde{\mathbf{w}})$ from Lemma 1 and using the property that $\mathbb{E}[(X - \mathbb{E}[X])^2] \leq \mathbb{E}[X^2]$ we get

$$\begin{aligned}\mathbb{E}[\|\mathbf{v}_t - \partial f(\mathbf{w}^*))\|^2] \leq & 2\mathbb{E}[\|G(\mathbf{w}_{t-1}, z_{i_t}) - G(\mathbf{w}^*, z_{i_t})\|^2] \\ & + 2\mathbb{E}[\|G(\mathbf{w}^*, z_{i_t}) - G(\tilde{\mathbf{w}}, z_{i_t})\|^2]\end{aligned}$$

From Lemma 2, we have $\|G(\mathbf{w}_{t-1}, z_{i_t}) - G(\mathbf{w}^*, z_{i_t})\| \leq 8M^2\|\mathbf{w}_{t-1} - \mathbf{w}^*\|$ and $\|G(\mathbf{w}^*, z_{i_t}) - G(\tilde{\mathbf{w}}, z_{i_t})\| \leq 8M^2\|\tilde{\mathbf{w}} - \mathbf{w}^*\|$. Using this, we can upper bound the variance of gradient step as:

$$\mathbb{E}[\|\mathbf{v}_t - \partial f(\mathbf{w}^*)]\|^2] \leq 2(8M^2)^2\|\mathbf{w}_{t-1} - \mathbf{w}^*\|^2 + 2(8M^2)^2\|\tilde{\mathbf{w}} - \mathbf{w}^*\|^2 \quad (3)$$

We have the desired result.

We now present the lemma that gives the bound on the variance of modified gradient $\mathbf{v}_t$.

Lemma 5. *Consider* VRSPAM *(Algorithm 1), then the variance of the* $\mathbf{v}_t$ *is upper bounded as:*

$$\mathbb{E}[\|\mathbf{v}_t - \partial f(\mathbf{w}_{t-1})\|^2] \leq 4(8M^2)^2\|\mathbf{w}_{t-1} - \mathbf{w}^*\|^2 + 2(8M^2)^2\|\tilde{\mathbf{w}} - \mathbf{w}^*\|^2$$

Proof.

$$\begin{aligned}\mathbb{E}[\|\mathbf{v}_t - \partial f(\mathbf{w}_{t-1})\|^2] &\leq 2\mathbb{E}[\|\mathbf{v}_t - \partial f(\mathbf{w}^*)\|^2] + 2\mathbb{E}[\|\partial f(\mathbf{w}^*) - \partial f(\mathbf{w}_{t-1})]\|^2] \\ &\leq 2(8M^2)^2\|\mathbf{w}_{t-1} - \mathbf{w}^*\|^2 + 2(8M^2)^2\|\tilde{\mathbf{w}} - \mathbf{w}^*\|^2 \\ &+ 2\mathbb{E}[\|G(\mathbf{w}_{t-1}, z_{i_t}) - G(\mathbf{w}^*, z_{i_t})\|^2] \\ &\leq 4(8M^2)^2\|\mathbf{w}_{t-1} - \mathbf{w}^*\|^2 + 2(8M^2)^2\|\tilde{\mathbf{w}} - \mathbf{w}^*\|^2\end{aligned}$$

where the second inequality uses Lemma 4 and last inequality uses Lemma 2.

At convergence, $\tilde{\mathbf{w}} = \mathbf{w}^*$ and $\mathbf{w}_t = \mathbf{w}^*$. Thus, the variance of the updates are bounded and go to zero as the algorithm converges whereas in the case of the SPAM algorithm, the variance of the gradient does not go to zero (which is a characteristic of a stochastic gradient descent based algorithm). We now present the proof of Theorem 1 using the above lemmas.

4.2 Proof of Theorem 1

From the first order optimality condition, we can directly write

$$\mathbf{w}^* = \text{prox}_{\eta\Omega}(\mathbf{w}^* - \eta\partial f(\mathbf{w}^*))$$

Using the above we can write

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 = \|\,\text{prox}_{\eta\Omega}(\hat{\mathbf{w}}_{t+1}) - \text{prox}_{\eta\Omega}(\mathbf{w}^* - \eta\partial f(\mathbf{w}^*))\|^2$$

Using Proposition 23.11 from [2], we have $\text{prox}_{\eta\Omega}$ is $(1+\eta\beta)$-cocoercieve and for any $\mathbf{u}$ and $\mathbf{w}$ using Cauchy Schwartz we can get the following inequality

$$\|\,\text{prox}_{\eta\Omega}(\mathbf{u}) - \text{prox}_{\eta\Omega}(\mathbf{w})\| \leq \frac{1}{1+\eta\beta}\|\mathbf{u} - \mathbf{w}\|$$

From above we get

$$\begin{aligned}\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 &\leq \frac{1}{(1+\eta\beta)^2}\|(\hat{\mathbf{w}}_{t+1}) - (\mathbf{w}^* - \eta\partial f(\mathbf{w}^*))\|^2 \\ &\leq \frac{1}{(1+\eta\beta)^2}\|(\mathbf{w}_t - \mathbf{w}^*) - \eta(G(\mathbf{w}_t, z_{i_{t+1}}) - G(\tilde{\mathbf{w}}, z_{i_{t+1}}) + \tilde{\boldsymbol{\mu}} - \partial f(\mathbf{w}^*))\|^2\end{aligned}$$

Taking expectation on both sides we get

$$\begin{aligned}\mathbb{E}\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \leq \frac{1}{(1+\eta\beta)^2}\big(&\eta^2\mathbb{E}[\|G(\mathbf{w}_t, z_{i_{t+1}}) - G(\tilde{\mathbf{w}}, z_{i_{t+1}}) + \tilde{\boldsymbol{\mu}} - \partial f(\mathbf{w}^*))\|^2] \\ + \mathbb{E}[\|\mathbf{w}_t - \mathbf{w}^*\|^2] - 2\eta\mathbb{E}[\langle\mathbf{w}_t - \mathbf{w}^*, &G(\mathbf{w}_t, z_{i_{t+1}}) - G(\tilde{\mathbf{w}}, z_{i_{t+1}}) + \tilde{\boldsymbol{\mu}} - \partial f(\mathbf{w}^*)\rangle]\big)\end{aligned} \tag{4}$$

Now, we first bound the last term $T = \mathbb{E}[\langle \mathbf{w}_t - \mathbf{w}^*, G(\mathbf{w}_t, z_{i_{t+1}}) - G(\tilde{\mathbf{w}}, z_{i_{t+1}}) + \tilde{\boldsymbol{\mu}} - \partial f(\mathbf{w}^*)\rangle]$ in Eq. 4. Using Lemma 1 we can write

$$\begin{aligned} T &= \mathbb{E}[\langle \mathbf{w}_t - \mathbf{w}^*, \mathbb{E}_{z_{t+1}}[G(\mathbf{w}_{t-1}, z_{i_{t+1}})] - \mathbb{E}_{z_{t+1}}[G(\tilde{\mathbf{w}}, z_{i_{t+1}})] + \tilde{\boldsymbol{\mu}} - \partial f(\mathbf{w}^*)\rangle] \\ &= \mathbb{E}[\langle \mathbf{w}_t - \mathbf{w}^*, \mathbb{E}_{z_{t+1}}[G(\mathbf{w}_t, z_{i_{t+1}})] - \partial f(\mathbf{w}^*)\rangle] \\ &= \mathbb{E}[\langle \mathbf{w}_t - \mathbf{w}^*, \partial f(\mathbf{w}_t) - \partial f(\mathbf{w}^*)\rangle] \\ &\geq 0 \end{aligned}$$

Now, $\mathbb{E}\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2$ can be bounded by using above bound and Lemma 4 as below

$$\begin{aligned} \mathbb{E}\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 &\leq \frac{1}{(1+\eta\beta)^2}(\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}^*\|^2] \\ &\quad + 2(8M^2)^2\eta^2(\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}^*\|^2] + \mathbb{E}[\|\tilde{\mathbf{w}} - \mathbf{w}^*\|^2])) \\ &\leq \frac{1 + 128M^4\eta^2}{(1+\eta\beta)^2}\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}^*\|^2] + \frac{128M^4\eta^2}{(1+\eta\beta)^2}\mathbb{E}[\|\tilde{\mathbf{w}} - \mathbf{w}^*\|^2] \end{aligned}$$

Let $C = \frac{1+128M^4\eta^2}{(1+\eta\beta)^2}$ and $D = \frac{128M^4\eta^2}{(1+\eta\beta)^2}$, then after m iterations $\mathbf{w}_t = \tilde{\mathbf{w}}_s$ and $\mathbf{w_0} = \tilde{\mathbf{w}}_{s-1}$. Substituting this in the above inequality, we get

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_s - \mathbf{w}^*\|^2 &\leq C^m\big(\mathbb{E}\|\tilde{\mathbf{w}}_{s-1} - \mathbf{w}^*\|^2 + \sum_{i=0}^{m-1}\frac{D}{C^i}\mathbb{E}\|\tilde{\mathbf{w}}_{s-1} - \mathbf{w}^*\|^2\big) \\ &\leq \big(C^m + \sum_{i=0}^{m-1}\frac{DC^m}{C^i}\big)\mathbb{E}\|\tilde{\mathbf{w}}_{s-1} - \mathbf{w}^*\|^2 \\ &\leq \big(C^m + DC^m\frac{1-(1/C^m)}{1-(1/C)}\big)\mathbb{E}\|\tilde{\mathbf{w}}_{s-1} - \mathbf{w}^*\|^2 \\ &\leq \big(C^m + DC\frac{C^m-1}{C-1}\big)\mathbb{E}\|\tilde{\mathbf{w}}_{s-1} - \mathbf{w}^*\|^2 \\ &\leq \alpha\mathbb{E}\|\tilde{\mathbf{w}}_{s-1} - \mathbf{w}^*\|^2 \end{aligned}$$

where $\alpha = C^m + DC\frac{C^m-1}{C-1}$ is the decay parameter, and $\alpha < 1$ by using Lemma 3. After s steps in outer loop of Algorithm 1, we get $\mathbb{E}\|\tilde{\mathbf{w}}_s - \mathbf{w}^*\|^2 \leq \alpha^s\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2$ where $\alpha < 1$. Hence, we get geometric convergence of α^s which is much stronger than the $\mathcal{O}(\frac{1}{t})$ convergence obtained in [17]. In the next section we derive the time complexity of the algorithm and investigate dependence of α on the problem parameters.

4.3 Complexity Analysis

To have $\mathbb{E}\|\tilde{\mathbf{w}}_s - \mathbf{w}^*\|^2 \leq \epsilon$, the number of iterations s must satisfy:

$$s \geq \frac{1}{\log\frac{1}{\alpha}}\log\frac{\mathbb{E}\|\mathbf{w}_0 - \mathbf{w}^*\|^2}{\epsilon}$$

At each stage, the number of gradient evaluations are $n + 2m$ where n is the number of samples and m is the iterations in the inner loop and the complexity is $\mathcal{O}(n+m)(\log(\frac{1}{\epsilon}))$ i.e. Algorithm 1 takes $\mathcal{O}(n+m)(\log(\frac{1}{\epsilon}))$ gradient complexity to achieve accuracy of ϵ. Here, the complexity is dependent on M and β as m itself is dependent on M and β.

Now we find the dependence of α and m on M and β. Let $\eta = \frac{\theta\beta}{128M^4}$ where $0 < \theta < 1$. Then,

$$\begin{aligned} C &= \frac{1 + 128M^4\eta^2}{(1+\eta\beta)^2} = \frac{1 + \frac{\theta^2\beta^2}{128M^4}}{(1 + \frac{\theta\beta^2}{128M^4})^2} \\ &< \frac{1 + \frac{\theta\beta^2}{128M^4}}{(1 + \frac{\theta\beta^2}{128M^4})^2} \\ &= \frac{1}{(1 + \frac{\theta\beta^2}{128M^4})} \\ &= E \end{aligned}$$

Therefore, $D = \theta(E - E^2)$ and $DC < \theta E^2(1-E)$, and using the above equations we can simplify α as

$$\begin{aligned} \alpha &= C^m + DC\frac{1-C^m}{1-C} \\ &< C^m + \theta E^2(1-E)\frac{1-C^m}{1-C} \\ &< C^m + \theta E^2(1-C^m) \quad \because \frac{1-E}{1-C} < 1 \\ &= \theta E^2 + C^m - \theta E^2\, C^m \end{aligned}$$

In the above equation, only $C^m - \theta E^2 C^m$ depends on m. If we choose m to be sufficiently large then $\alpha = \theta E^2$. An important thing to note here is that $\theta E < C < E$, now if we choose $m \approx 2\frac{\log\theta}{\log E}$ then $\alpha \approx 2\theta E^2$. Thus, the time complexity of the algorithm is

$$\mathcal{O}(n + 2\frac{\log\theta}{\log E})(\log(\frac{1}{\epsilon})) \quad \text{when} \quad m = \Theta(\frac{\log\theta}{\log E}).$$

As the order has inverse dependency on $\log E = \log\frac{128M^4}{128M^4+\theta\beta^2}$, increase in M will result in increase in number of iterations i.e. as the maximum norm of training samples is increased, larger m is required to reach ϵ accuracy.

Comparison of Time Complexities: Now let us compare the time complexities of our algorithm with that of the SPAM algorithm. First, we derive the time complexity of SPAM. We will use Theorem 3 from [17] which states that SPAM achieves the following:

$$\mathbb{E}[\|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2] \leq \frac{t_0}{T}\mathbb{E}[\|\mathbf{w}_{t_0} - \mathbf{w}^*\|^2] + c\frac{\log T}{T}$$

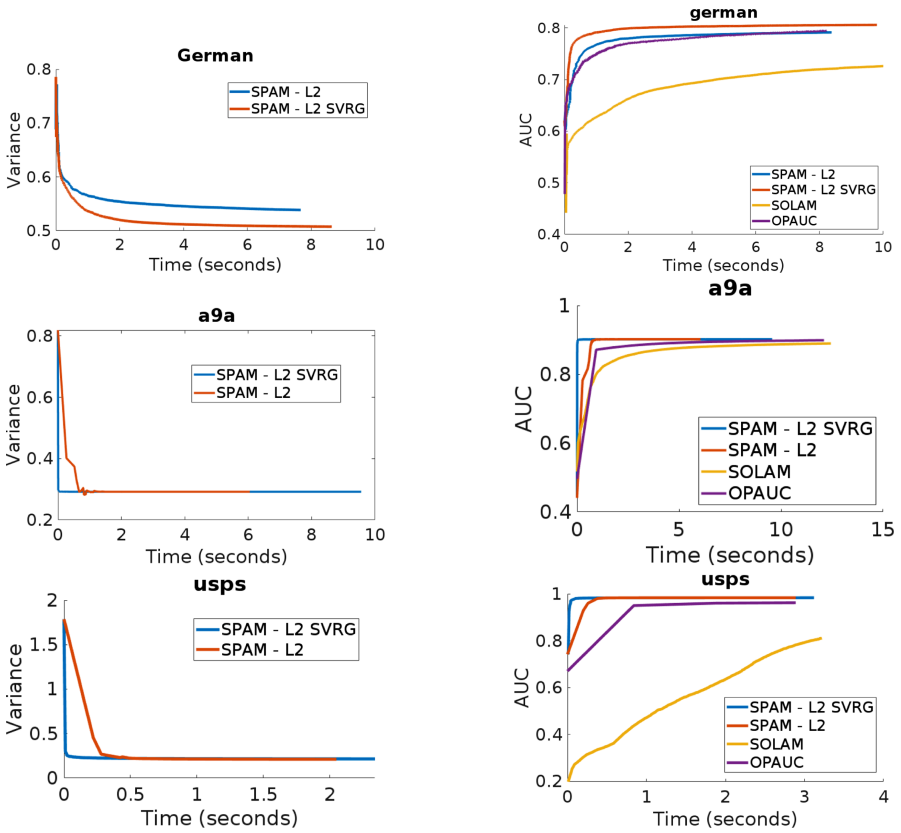

Fig. 1. The left column shows that VRSPAM (SPAM-L2-SVRG) has lower variance than SPAM-L2 across different datasets. The right column shows VRSPAM (SPAM-L2-SVRG) converges faster and performs better than existing algorithms on AUC maximization

where $t_0 = \max\big(2, \lceil 1 + \frac{(128M^4+\beta^2)^2}{128M^4\beta^2} \rceil\big)$, T is the number of iterations and c is a constant. Using the averaging scheme developed by [15], the following can be obtained:

$$\mathbb{E}[\|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2] \leq \frac{t_0}{T}\mathbb{E}[\|\mathbf{w}_{t_0} - \mathbf{w}^*\|^2] \tag{5}$$

where

$$\mathbb{E}[\|\mathbf{w}_{t_0} - \mathbf{w}^*\|^2] \leq \frac{2\sigma_*^2}{\tilde{C}^2_{\beta,M}} + \exp\Big(\frac{128M^4}{\tilde{C}^2_{\beta,M}}\Big) = F,$$

$$\tilde{C}^2_{\beta,M} = \frac{\beta}{(1 + \frac{\beta^2}{128M^4})^2} \quad \text{and} \quad \mathbb{E}[\|G(\mathbf{w}^*; z) - \partial f(\mathbf{w}^*)\|^2] = \sigma_*^2.$$

Using Eq. 5, the time complexity of the SPAM algorithm can be written as $\mathcal{O}(\frac{t_0 F}{\epsilon})$ i.e. SPAM takes $\mathcal{O}(\frac{t_0 F}{\epsilon})$ iterations to achieve ϵ accuracy. Thus, SPAM

has lower per iteration complexity but slower convergence rate when compared to VRSPAM. In other words, VRSPAM will take less time to get a good approximation of the solution.

Table 1. Datasets used for evaluating VRSPAM and the different state-of-the-art algorithms for AUC maximization.

N	Name	Instances	Features
1	DIABETES	768	8
2	GERMAN	1000	24
3	SPLICE	3,175	60
4	USPS	9,298	256
5	LETTER	20,000	16

Data	Name	Instances	Features
6	A9A	32,561	123
7	W8A	64,700	300
8	MNIST	60,000	780
9	ACOUSTIC	78,823	50
10	IJCNN1	141,691	22

5 Experiment

In this section, we empirically compare the performance of VRSPAM with other existing algorithms for AUC maximization, on several standard benchmarks. We use the following two variants of our proposed algorithm based on the regularizer used:

1. VRSPAM $-\ L^2 : \Omega(\mathbf{w}) = \frac{\beta}{2}\|\mathbf{w}\|^2$ (Frobenius Norm Regularizer)
2. VRSPAM $-\ NET\ :\ \Omega(\mathbf{w}) = \frac{\beta}{2}\|\mathbf{w}\|_2^2 + \beta_1\|\mathbf{w}\|_1$ (Elastic Net Regularizer [27]). The proximal step for elastic net is given as $\arg\min_{\mathbf{w}}\{\frac{1}{2}\|\mathbf{w} - \frac{\hat{\mathbf{w}}_{t+1}}{\eta_t\beta+1}\|^2 + \frac{\eta_t\beta_1}{\eta_t\beta+1}\|\mathbf{w}\|_1\}$

VRSPAM is compared with several baselines: SPAM, SOLAM [24] and one-pass AUC optimization algorithm (OPAUC) [9], which are state-of-the-art methods for AUC maximization. SOLAM was modified to have the Frobenius Norm Regularizer (as in [17]). VRSPAM is compared against OPAUC with the least square loss.

Table 2. Comparison of AUC values (mean±std) achieved by the different algorithms on the test data of the different datasets described in Table 1.

N	VRSPAM-L^2	VRSPAM-NET	SPAM-L^2	SPAM-NET	SOLAM	OPAUC
1	.8299±.0323	**.8305±.0319**	.8272±.0277	.8085±.0431	.8128±.0304	.8309±.0350
2	.7902±0386	.7845±.0398	.7942±.0388	.7937±.0386	.7778±.0373	**.7978±.0347**
3	.9640±.0156	**.9699±.0139**	.9263±.0091	.9267±.0090	.9246±.0087	.9232±.0099
4	**.8552±.006**	.8549±.0059	.8542±.0388	.8537±.0386	.8395±.0061	.8114±.0065
5	.9834±.0023	.9804±.0032	**.9868±.0032**	.9855±.0029	.9822±.0036	.9620±.0040
6	**.9003±.0045**	.8981±.0046	.8998±.0046	.8980±.0047	.8966±.0043	.9002±.0047
7	**.9876±.0008**	.9787±.0013	.9682±.0020	.9604±.0020	.9817±.0015	.9633±.0035
8	**.9465±.0014**	.9351±.0014	.9254±.0025	.9132±.0026	.9118±.0029	.9242±.0021
9	.8093±.0033	.8052±.033	.8120±.0030	.8109±.0028	.8099±.0036	**.8192±.0032**
10	**.9750±.001**	.9745±.002	.9174±.0024	.9155±.0024	.9129±.0030	.9269±.0021

All datasets are publicly available from [4] and [8]. Some of the datasets, like MNIST, are multiclass, and we convert them to binary labels by numbering the classes and assigning all the even labels to one class and all the odd labels to another. The results are the mean AUC score and standard deviation of 20 runs on each dataset. All the datasets were randomly divided into training and test splits with 80% and 20% of the data. The parameters $\beta \in 10^{[-5:5]}$ and $\beta_1 \in 10^{[-5:5]}$ for VRSPAM $- L^2$ and VRSPAM $- NET$ are chosen by a 5 fold cross-validation on the training set. All the code is implemented in MATLAB. We measured the algorithm's computational time using an Intel $i7$ CPU with a clock speed of 3538 MHz.

5.1 VRSPAM Has Lower Variance

Theoretically, we derived that VRSPAM has lower variance than the baseline SPAM algorithm. Here, we see empirically this holds across the different datasets. In the left column of Fig. 1, we show the variance of the VRSPAM update $(\mathbf{v}_t)$ in comparison with the variance of SPAM update $(G(\mathbf{w}_{t-1}, z_{i_{t-1}}))$. We observe that the variance of VRSPAM is lower than the variance of SPAM and decreases to the minimum value faster, which is in line with Theorem 1.

5.2 VRSPAM Has Faster Convergence

Theoretically, we derived that VRSPAM converges faster than the baseline SPAM algorithm. Here, we see empirically this holds across the different datasets. In the right column of Fig. 1, we show the performance of VRSPAM compared to existing methods for AUC maximization. We observe that VRSPAM converges to the maximum value faster than the other methods, and in some cases, this maximum value itself is higher for VRSPAM.

We found that the best results were obtained when the initial weights of VRSPAM were set to be the output generated by SPAM after one iteration, which happens to be standard practice in related problems in optimization [13]. Table 2 summarizes the results of the performance of different algorithms as measured by the AUC metric, across different datasets. AUC values for SPAM-L^2, SPAM-NET, SOLAM and OPAUC were taken from [17]. It is seen that in almost all the datasets, one of the two versions of VRSPAM has the best performance and this gain is consistent across multiple runs, as seen by the standard error. This shows that under finite computational time, VRSPAM is able to converge to the global optimum faster than the other algorithms.

6 Conclusion

In this paper, we propose a variance reduced stochastic proximal algorithm for AUC maximization (VRSPAM). We theoretically analyze the proposed algorithm and derive a much faster convergence rate of $\mathcal{O}(\alpha^t)$ where $\alpha < 1$ (linear convergence rate), improving upon state-of-the-art methods [17] which have

a convergence rate of $\mathcal{O}(\frac{1}{t})$ (sub-linear convergence rate), for strongly convex objective functions with per iteration complexity of one data-point. We gave a theoretical analysis of this and showed empirically VRSPAM converges faster than other methods for AUC maximization.

For future work, it will be interesting to explore if other algorithms used to accelerate SGD can be used in this setting and if they lead to even faster convergence. It is also interesting to apply the proposed methods in practice to non-decomposable performance measures other than AUC. It would be interesting to extend the analysis to a non-convex and non-smooth regularizer using method presented in [23].

References

1. Agarwal, S.: Surrogate regret bounds for the area under the roc curve via strongly proper losses. In: Conference on Learning Theory, pp. 338–353 (2013)
2. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. CBM, Springer, Cham (2017). https://doi.org/10.1007/978-3-319-48311-5
3. Beck, A., Teboulle, M.: A fast iterative shrinkage-threshold algorithm for linear inverse problems. Technion-Israel Institute of Technology, Technical Report (2008)
4. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) **2**(3), 27 (2011)
5. Clémençon, S., Lugosi, G., Vayatis, N., et al.: Ranking and empirical minimization of u-statistics. Ann. Statist. **36**(2), 844–874 (2008)
6. Elkan, C.: The foundations of cost-sensitive learning. In: International Joint Conference on Artificial Intelligence, vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)
7. Fawcett, T.: An introduction to roc analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006)
8. Frank, A., Asuncion, A.: UCI machine learning repository [http://archive.ics.uci.edu/ml]. University of California, Irvine. School of Information and Computer Science **213**, 2 (2010)
9. Gao, W., Jin, R., Zhu, S., Zhou, Z.H.: One-pass AUC optimization. In: International Conference on Machine Learning, pp. 906–914 (2013)
10. Gao, W., Zhou, Z.H.: On the consistency of AUC pairwise optimization. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
11. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology **143**(1), 29–36 (1982)
12. Herschtal, A., Raskutti, B.: Optimising area under the roc curve using gradient descent. In: Proceedings of the Twenty-First International Conference on Machine Learning, p. 49. ACM (2004)
13. Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: Advances in Neural Information Processing Systems, pp. 315–323 (2013)
14. Kar, P., Sriperumbudur, B.K., Jain, P., Karnick, H.C.: On the generalization ability of online learning algorithms for pairwise loss functions. In: Proceedings of the 30th International Conference on International Conference on Machine Learning, vol. 28, pp. III-441. JMLR. org (2013)

15. Lacoste-Julien, S., Schmidt, M., Bach, F.: A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. arXiv preprint arXiv:1212.2002 (2012)
16. Narasimhan, H., Agarwal, S.: Support vector algorithms for optimizing the partial area under the roc curve. Neural Comput. **29**(7), 1919–1963 (2017)
17. Natole, M., Ying, Y., Lyu, S.: Stochastic proximal algorithms for AUC maximization. In: International Conference on Machine Learning, pp. 3707–3716 (2018)
18. Orabona, F.: Simultaneous model selection and optimization through parameter-free stochastic learning. In: Advances in Neural Information Processing Systems, pp. 1116–1124 (2014)
19. Roux, N.L., Schmidt, M., Bach, F.R.: A stochastic gradient method with an exponential convergence _rate for finite training sets. In: Advances in Neural Information Processing Systems, pp. 2663–2671 (2012)
20. Shalev-Shwartz, S., Zhang, T.: Stochastic dual coordinate ascent methods for regularized loss minimization. J. Mach. Learn. Res. **14**(Feb), 567–599 (2013)
21. Shalev-Shwartz, S., et al.: Online learning and online convex optimization. Found. Trends® Mach. Learn. **4**(2), 107–194 (2012)
22. Xiao, L., Zhang, T.: A proximal stochastic gradient method with progressive variance reduction. SIAM J. Optim. **24**(4), 2057–2075 (2014)
23. Xu, Y., Qi, Q., Lin, Q., Jin, R., Yang, T.: Stochastic optimization for dc functions and non-smooth non-convex regularizers with non-asymptotic convergence. In: International Conference on Machine Learning, pp. 6942–6951 (2019)
24. Ying, Y., Wen, L., Lyu, S.: Stochastic online AUC maximization. In: Advances in Neural Information Processing Systems, pp. 451–459 (2016)
25. Zhang, X., Saha, A., Vishwanathan, S.: Smoothing multivariate performance measures. J. Mach. Learn. Res. **13**(Dec), 3623–3680 (2012)
26. Zhao, P., Hoi, S.C., Jin, R., Yang, T.: Online AUC maximization. In: Proceedings of the 28th International Conference on International Conference on Machine Learning, pp. 233–240. Omnipress (2011)
27. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Statist. Soc. B Statist. Methodol. **67**(2), 301–320 (2005)