Scalable and Efficient Pairwise Learning to Achieve Statistical Accuracy

Bin Gu[1], **Zhouyuan Huo**[2], **Heng Huang**[2,1*]
[1]JDDGlobal.com, [2]Department of Electrical & Computer Engineering, University of Pittsburgh, USA
gubin3@jd.com, zhouyuan.huo@pitt.edu, heng.huang@pitt.edu

Abstract

Pairwise learning is an important learning topic in the machine learning community, where the loss function involves pairs of samples (*e.g.*, AUC maximization and metric learning). Existing pairwise learning algorithms do not perform well in the generality, scalability and efficiency simultaneously. To address these challenging problems, in this paper, we first analyze the relationship between the statistical accuracy and the regularized empire risk for pairwise loss. Based on the relationship, we propose a scalable and efficient adaptive doubly stochastic gradient algorithm (AdaDSG) for generalized regularized pairwise learning problems. More importantly, we prove that the overall computational cost of AdaDSG is $\mathcal{O}(n)$ to achieve the statistical accuracy on the full training set with the size of n, which is the best theoretical result for pairwise learning to the best of our knowledge. The experimental results on a variety of real-world datasets not only confirm the effectiveness of our AdaDSG algorithm, but also show that AdaDSG has significantly better scalability and efficiency than the existing pairwise learning algorithms.

Introduction

Many machine learning problems, such as AUC maximization (Zhao et al., 2011; Gao et al., 2013) or equivalently bipartite ranking (Agarwal and Niyogi, 2009; Rejchel, 2012), metric learning (Jin, Wang, and Zhou, 2009; Weinberger and Saul, 2009; Ying and Li, 2012) and multiple kernel learning (Kumar et al., 2012), consider the pairwise loss function on a pair of samples (x, y) and (x', y') of the form of $L(f, (x, y), (x', y'))$. For example, Gao et al. (2013) considered the least square pairwise loss function $(1 - (f(x) - f(x')))^2$ for AUC maximization, where y and y' are with different labels. This important learning scenario is called as pairwise learning. The aim of pairwise learning is to find a hypothesis function minimizing the expected risk $\mathbb{E}_{(x,y)}\mathbb{E}_{(x',y')}L(f, (x, y), (x', y'))$.

The scalability and efficiency have been the notorious bottlenecks of pairwise learning. Traditional univariate loss functions only depend on one sample. The problem size for traditional machine learning problems grows linearly in the size of samples. However, as mentioned above, pairwise loss functions depend on pair of samples. Thus, the pairwise learning algorithms need to handle the challenge raised by the big volume of data samples in the sense that, the size of pairs of samples grows quadratically in term of the size of samples. If the training set has n samples, we will have n^2 possible pairs of samples which make it challenging to design a scalable and efficient pairwise learning algorithm.

Existing pairwise learning algorithms have mainly utilized the techniques of online learning and stochastic optimization to address the challenge of the quadratic growth of the size of sample pairs. Specifically, Lin et al. (2017) used the typical online learning framework (Cesa-Bianchi, Conconi, and Gentile, 2004) to implement pairwise learning whose space and time complexities are $\mathcal{O}(Td)$ and $\mathcal{O}(T^2d)$ respectively, where d is the dimensionality and T is the iteration number. Kar et al. (2013) proposed an improved online algorithm for pairwise learning by utilizing a buffer of a fixed size s to update the gradients, whose space and time complexities are $\mathcal{O}(sd)$ and $\mathcal{O}(sdT)$ respectively. Boissier et al. (2016) introduced an improved online algorithm for linear pairwise learning with the space and time complexities as $\mathcal{O}(d^2)$ and $\mathcal{O}(Td^2)$ respectively, by incrementally updating the gradients. Gao et al. (2013) used the similar strategy to implement AUC maximization. Ying, Wen, and Lyu (2016) reformulated the AUC maximization problem as a saddle point problem, and proposed a stochastic optimization algorithm with the space and time complexities as $\mathcal{O}(d)$ and $\mathcal{O}(Td)$ respectively. We summarize these representative pairwise learning algorithms in Table 1.

We compare the pairwise learning algorithms from three points of view, *i.e.*, *generality*, *complexity*, and *convergence rate*. For *generality*, we consider the generalities *w.r.t.* solved problems, pairwise loss functions, and hypothesis functions. For *complexity*, we consider the complexities *w.r.t.* space and time. From Table 1, we find that the online algorithm of (Kar et al., 2013) has the best generality and convergence rate in the existing pairwise learning algorithms. However, its space and time complexities are related to a buffer size s and achieving better generalization performance requires a larger s. Thus, the online algorithm of (Kar et al., 2013) is still not scalable and efficient enough. Although the SOLAM algorithm (Ying, Wen, and Lyu, 2016) has the best

[*]Corresponding Author. Z. Huo, H. Huang were partially supported by U.S. NSF IIS 1836945, IIS 1836938, DBI 1836866, IIS 1845666, IIS 1852606, IIS 1838627, IIS 1837956.

Table 1: Representative pairwise learning algorithms. (PL and FCN are the abbreviations of pairwise loss and function, respectively. T is the iteration number, d is the dimensionality and s is the buffer size.)

Algorithm	Reference	Generalization			Complexity		Convergence rate
		Problems	Loss FCN	Hypothesis FCN	Space	Time	
Online	Lin et al. (2017)	General PL	Square	Kernel	$\mathcal{O}(Td)$	$\mathcal{O}(T^2d)$	$\mathcal{O}(T^{-\frac{1}{4}}\ln T)$
Online	Kar et al. (2013)	General PL	General	General	$\mathcal{O}(sd)$	$\mathcal{O}(sTd)$	$\mathcal{O}(\frac{1}{T})$
Online	Boissier et al. (2016)	General PL	Square	Linear	$\mathcal{O}(d^2)$	$\mathcal{O}(Td^2)$	$\mathcal{O}(\frac{\log^2 T}{T})$
OPAUC	Gao et al. (2013)	AUC	Square	Linear	$\mathcal{O}(d^2)$	$\mathcal{O}(Td^2)$	$\mathcal{O}(\frac{1}{\sqrt{T}})$
SOLAM	Ying and Li (2012)	AUC	Square	Linear	$\mathcal{O}(d)$	$\mathcal{O}(Td)$	$\mathcal{O}(\frac{(\ln T)^{1.5}}{\sqrt{T}})$
AdaDSG	Our	General PL	General	General	$\mathcal{O}(d)$	$\mathcal{O}(Td)$	At least $\mathcal{O}(\frac{1}{T})$

space and time complexities, it works only for AUC maximization, and has a poor convergence rate. To the best of our knowledge, the existing pairwise learning algorithms do not perform well in the generality, complexities, and convergence rate simultaneously. The scalability and efficiency are still the bottlenecks of existing pairwise learning algorithms.

To address these challenges, in this paper, we first analyze the relationship between the statistical accuracy and the regularized empire risk for pairwise loss. Based on the relationship, we propose a scalable and efficient adaptive doubly stochastic gradient algorithm (AdaDSG) for regularized pairwise learning problems following the adaptive sample size scheme. More importantly, we prove that the overall computational cost of AdaDSG is $\mathcal{O}(n)$ to achieve the statistical accuracy on the training set with the size of n, which is the best theoretical result for pairwise learning to the best of our knowledge. The experiments on the application of the AUC maximization are conducted to validate our AdaDSG algorithm. The experimental results on a variety of real-world datasets not only confirm the effectiveness of our AdaDSG algorithm, but also show that AdaDSG has significantly better scalability and efficiency than the existing pairwise learning algorithms.

Contributions. The main contributions of this paper are summarized as follows:

1. The existing adaptive sample size algorithms only focus on the full or *singly* stochastic gradient algorithms for *univariate* loss functions. Differently, our AdaDSG algorithm is the first adaptive sample size algorithm working on the *doubly* stochastic gradient algorithm for *pairwise* loss functions.
2. The existing adaptive sample size algorithms require a *strong* assumption of convergence rate (*i.e.*, *linear* or *quadratic*) *w.r.t.* the full or stochastic gradient algorithms. However, to the best of our knowledge, our AdaDSG is the first adaptive sample size algorithm working on a *weaker* assumption of convergence rate (*i.e. sublinear*) for the doubly stochastic gradient algorithm.
3. The convergence rate of our AdaDSG algorithm is at least $\mathcal{O}(\frac{1}{T})$. More importantly, we prove that the overall computational cost of AdaDSG is $\mathcal{O}(n)$ to achieve the statistical accuracy on the full training set with the size of n, which is the best theoretical result for pairwise learning to the best of our knowledge.

Organization. We organize the rest of paper as follows. Firstly, we present several related works. Secondly, we present the generalized pairwise learning problem considered in this paper. Thirdly, we analyze the statistical accuracy in pairwise learning problems. Fourthly, we propose our AdaDSG algorithm and give its complexity analysis. Fifthly, we show the experimental results of AUC maximization on a variety of datasets. Finally, we conclude the paper.

Related Work

Essentially, our AdaDSG algorithm is an adaptive doubly stochastic gradient algorithm following the adaptive sample size scheme. In this section, we first give a brief review of doubly stochastic optimization algorithms, and then give a brief review of adaptive sample size algorithms.

Doubly Stochastic Optimization

According to how many random events occur per iteration, stochastic optimization algorithms can be divided into the singly stochastic approach, the doubly stochastic approach and others. Normally the sample space (*i.e.*, the union of all possible random events) of stochastic optimization algorithms could be the set of samples or the set of coordinates. For example, the sample space of stochastic gradient descent algorithms (Defazio, Bach, and Lacoste-Julien, 2014; Johnson and Zhang, 2013) on the univariate loss functions is the set of samples. The sample space of stochastic coordinate descent algorithms (Bradley et al., 2011; Liu and Wright, 2015) is the set of coordinates. For traditional doubly stochastic optimization algorithms (Dai et al., 2014; Zhao et al., 2014; Gu, Huo, and Huang, 2018; Gu et al., 2018), the sample spaces are both the set of samples and the set of coordinates. Our AdaDSG algorithm considers the pairwise loss functions and repeats the two random events on the same sample space (*i.e.*, the set of samples). Thus, different to the traditional doubly stochastic optimization algorithms which have two different sample spaces, our adaptive doubly stochastic gradient algorithm has one sample space, *i.e.*, the set of samples.

Adaptive Sample Size Algorithms

There have been several adaptive sample size algorithms proposed to solve the (regularized) empirical risk problems of traditional univariate loss. Specifically, Daneshmand, Lucchi, and Hofmann (2016) proposed the adaptive sample size scheme for the empirical risk problems of traditional univariate loss on SAGA algorithm (DynaSAGA) (Defazio, Bach, and Lacoste-Julien, 2014). Later, Mokhtari et al. (2016); Eisen, Mokhtari, and Ribeiro (2018) considered the regularized empirical risk problems of traditional univariate loss, and extended the adaptive sample size scheme to Newton's method (Boyd and Vandenberghe, 2004). Mokhtari and Ribeiro (2017) also considered the regularized empirical risk problems of traditional univariate loss, and extended the adaptive sample size scheme to accelerated gradient descent (Yu, 2013) and SVRG (Johnson and Zhang, 2013) algorithms. All these works proved that the computational complexities can be reduced to $\mathcal{O}(n^{\frac{5}{4}})$ or $\mathcal{O}(n)$ to reach the statistical accuracy on the full training set. We also summarize these representative adaptive sample size algorithms in Table 2. To sum up, existing adaptive sample size framework works only for the traditional univariate loss functions, where the (stochastic) gradient algorithms are with linear or quadratic convergence rate. However, our AdaDSG algorithm works for pairwise loss functions on a *weaker* assumption of convergence rate (*i.e.* sublinear) for the doubly stochastic gradient algorithm.

Generalized Pairwise Learning Problem

As mentioned previously, the ultimate goal of pairwise learning in theory is to find an optimal argument that minimizes the expected risk $\mathcal{L}(w)$ *w.r.t.* a pairwise loss function of the form of:

$$\begin{aligned} w^* &= \operatorname*{argmin}_{w\in\mathbb{R}^d} \mathcal{L}(w) \qquad (1) \\ &= \operatorname*{argmin}_{w\in\mathbb{R}^d} \mathbb{E}_{(x,y)}\mathbb{E}_{(x',y')} L(f_w,(x,y),(x',y')) \end{aligned}$$

where f_w is a hypothesis function with parameter w. However, due to the fact that the distribution of samples is unknown, it is challenging to minimize the expected risk $\mathcal{L}(w)$. In the real world applications of pairwise learning, instead of minimizing the expected risk $\mathcal{L}(w)$, we usually consider the empirical risk of pairwise loss function on a training set $\mathcal{S} = \{(x_i,y_i)\}_{i=1}^n$ as follows.

$$\mathcal{L}_n(w) = \frac{1}{n(n-1)} \sum_{i,j\in\mathcal{S},i\neq j} L(f_w,(x_i,y_i),(x_j,y_j)) \qquad (2)$$

Obviously, the problem (2) covers various pairwise learning problems, including AUC maximization (Gao et al., 2013) (equivalently called bipartite ranking (Rejchel, 2012)), metric learning (Jin, Wang, and Zhou, 2009), and multiple kernel learning (Kumar et al., 2012). Note that the pairwise loss function $L(f_w,(x_i,y_i),(x_j,y_j))$ is equal to zero for AUC maximization if $y_i = y_j$.

Although the sample size of the set $\mathcal{S}$ could be huge in the era of big data, it is still possible to overfit the training set if directly minimizing the empirical risk objective (2). To prevent overfitting, we add a regularization term $\lambda\|w\|^2$ to the empirical risk $\mathcal{L}_n(w)$, Thus, in this paper, we find an optimal argument that optimizes a regularized empire risk $R_n(w)$ as mentioned in (3), instead of the empirical risk $\mathcal{L}_n(w)$.

$$w_n^* = \operatorname*{argmin}_{w\in\mathbb{R}^d} R_n(w) = \operatorname*{argmin}_{w\in\mathbb{R}^d} \mathcal{L}_n(w) + \lambda\|w\|^2$$

To build the relationship between statistical accuracy and regularized empirical risk, we rewrite the regularization parameter λ in the formulation (3) as form of $\lambda = \frac{cV_n}{2}$, where $V_n = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ and c is a constant to control the regularization parameter $\frac{cV_n}{2}$. Thus, the formulation of regularized empire risk $R_n(w)$ can be reformulated as follows.

$$\begin{aligned} & R_n(w) \qquad (3) \\ = \;& \mathcal{L}_n(w) + \frac{cV_n}{2}\|w\|^2 = \frac{1}{n(n-1)} \cdot \\ & \sum_{i,j\in\mathcal{S},i\neq j} \left(\underbrace{L(f_w,(x_i,y_i),(x_j,y_j)) + \frac{cV_n}{2}\|w\|^2}_{F_{i,j}(w)} \right) \end{aligned}$$

where each function $F_{i,j} : \mathbb{R}^d \to \mathbb{R}$ is a smooth convex function. Note that, the regularization term $\frac{cV_n}{2}\|w\|^2$ not only avoids overfitting, but also ensures that the problem is strongly convex.

Statistical Accuracy in Pairwise Learning Problems

In this section, we first analyze the relationship between statistical accuracy and empirical risk $\mathcal{L}_n(w)$, then analyze the relationship between statistical accuracy and regularized empirical risk $R_n(w)$.

Statistical Accuracy and Empirical Risk

There have been several works to give the upper bounds of the difference between the expected risk $\mathcal{L}$ and the empirical risk $\mathcal{L}_n$ for AUC maximization (Agarwal et al., 2005), bipartite ranking (Agarwal and Niyogi, 2009), and metric learning (Cao, Guo, and Ying, 2016). Recently, Lei, Lin, and Tang (2018) provided a unified upper bound on the difference between the expected and empirical risks for pairwise learning. To make this paper self-contained, we provide this generalized upper bound as follows.

Theorem 1. *(Lei, Lin, and Tang, 2018) Given an i.i.d. training set $\mathcal{S} = \{(x_i,y_i)\}_{i=1}^n$ for pairwise learning, we have an upper bound on the difference between the expected risk $\mathcal{L}$ and the empirical risk $\mathcal{L}_n$ for all $w \in \mathbb{R}^d$ as follows:*

$$\mathbb{E}\left[\sup_{w\in\mathbb{R}^d} |\mathcal{L}_n(w) - \mathcal{L}(w)|\right] \leq V_n\,, \qquad (4)$$

where $V_n = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$.

According to Theorem 1, we have that the optimal values of the expected loss and empirical loss are within a V_n distance at least of each other. Based on Theorem 1, we conclude that there is no gain in improving the optimization

Table 2: Representative adaptive sample size algorithms. (CRS is the abbreviation of convergence rate of subsolver.)

Algorithm	Reference	Loss function	CRS	Complexity
DynaSAGA	Daneshmand, Lucchi, and Hofmann (2016)	Univariate	Linear	$\mathcal{O}(n)$
AdaNewton	Mokhtari et al. (2016)	Univariate	Quadratic	$\mathcal{O}(n)$
AdaAGD	Mokhtari and Ribeiro (2017)	Univariate	Linear	$\mathcal{O}(n^{\frac{5}{4}})$
AdaSVRG	Mokhtari and Ribeiro (2017)	Univariate	Linear	$\mathcal{O}(n)$
AdaDSG	Our	Pairwise	Sublinear	$\mathcal{O}(n)$

error of minimizing $\mathcal{L}_n$ beyond the constant V_n. In other words, if we find an approximate solution w_n such that the optimization error is bounded by $\mathcal{L}_n(w_n) - \mathcal{L}_n(w_n^\dagger) \leq V_n$, where $w_n^\dagger = \operatorname{argmin}_{w \in \mathbb{R}^d} \mathcal{L}_n(w)$, finding a more accurate solution to reduce the optimization error is not beneficial. This conclusion is confirmed by Theorem 2 (the detailed proof to Theorem 2 can be found in our Appendix).

Theorem 2. *Given an i.i.d. training set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$. Define w_n as a δ_n optimal solution of the risk $\mathcal{L}_n$ in expectation,* i.e., $\mathbb{E}\left[\mathcal{L}_n(w_n) - \mathcal{L}_n(w_n^\dagger)\right] \leq \delta_n$. *We have that*

$$\mathbb{E}\left[\mathcal{L}_n(w_n) - \mathcal{L}(w^*)\right] \leq \delta_n + 3V_n \,. \tag{5}$$

Thus, it is easy to see that V_n is an important theoretical quantity. In this paper, we define V_n as the statistical accuracy as follows.

Definition 1 (Statistical accuracy)**.** *The statistical accuracy on an i.i.d. training set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$ is defined as $V_n = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$.*

Remark 1 (Relationship between V_n and $\mathcal{L}_n$)**.** *According to Theorem 2 and Definition 1, we say that w_n solves the empirical risk problem in (2) within its statistical accuracy if it satisfies $\mathcal{L}_n(w_n) - \mathcal{L}_n(w_n^\dagger) \leq V_n$.*

Statistical Accuracy and Regularized Empirical Risk

Now, let's consider the training set $\mathcal{S}_m$ with m samples as a subset of the full i.i.d. training dataset $\mathcal{S}$, *i.e.*, $\mathcal{S}_m \subset \mathcal{S}$. First, we solve the problem corresponding to the set $\mathcal{S}_m$ such that the approximate solution w_m satisfies the condition $\mathbb{E}[R_m(w_m) - R_m(w_m^*)] \leq \delta_m$. Next, we consider another training subset $\mathcal{S}_{m'}$ such that $\mathcal{S}_{m'}$ contains the set $\mathcal{S}_m$, *i.e.*, $\mathcal{S}_m \subset \mathcal{S}_{m'} \subseteq \mathcal{S}$. Thirdly, we use w_m as an initial solution of the problem $R_{m'}$ and solve the subproblem related to the set $\mathcal{S}_{m'}$.

A key question for the above procedure is that how much accuracy is enough for solving the subproblem $\mathcal{S}_m$. To answer this question, we derive an upper bound on the expected suboptimality of the variable w_m *w.r.t.* the problem $R_{m'}$ in Theorem 3, which is built on the accuracy of w_m *w.r.t.* the previous problem R_m associated to the training set $\mathcal{S}_m$. The detailed proof can be found in our Appendix. Based on Theorem 3 and Remark 2, we conclude that there is no gain in solving the subproblem R_m beyond its statistical accuracy V_m, if $m' = 2m$.

Theorem 3. *Let $\mathcal{L}_m$ and $\mathcal{L}_{m'}$ denote the empirical risks on the sets $\mathcal{S}_m$ and $\mathcal{S}_{m'}$, respectively, where they are chosen such that $\mathcal{S}_m \subset \mathcal{S}_{m'}$. Further, define w_m as an δ_m optimal solution of the risk R_m in expectation,* i.e., $\mathbb{E}[R_m(w_m) - R_m(w_m^*)] \leq \delta_m$. *Moreover, recall w^* as the optimal solution of the expected risk $\mathcal{L}$ as defined in (1). We have that:*

$$\begin{aligned} & \mathbb{E}\left[R_{m'}(w_m) - R_{m'}(w_{m'}^*)\right] \qquad\qquad (6) \\ \leq \quad & \delta_m + 2(V_{m'} + V_m) + \frac{c(V_m - V_{m'})}{2}\left(\frac{4}{c} + \|w^*\|^2\right) \end{aligned}$$

Remark 2 (Relationship between V_m and R_m)**.** *According to Theorem 3, if setting $m' = 2m$, we have that*

$$\begin{aligned} & \mathbb{E}\left[R_{2m}(w_m) - R_{2m}(w_{2m}^*)\right] \qquad\qquad (7) \\ \leq \quad & \delta_m + \left(4 + \frac{c(1 - \frac{1}{\sqrt{2}})}{2}\|w^*\|^2\right) V_m \end{aligned}$$

The inequality (7) shows that there is no need to solve the subproblem R_m beyond its statistical accuracy V_m. Specifically, even if δ_m is zero, the expected sub-optimality will be of the order $\mathcal{O}(V_m)$, i.e., $\mathbb{E}\left[R_{2m}(w_m) - R_{2m}(w_{2m}^*)\right] \leq \mathcal{O}(V_m)$. *Based on the inequality (7), the required precision δ_m for solving the subproblem R_m should be order of $\mathcal{O}(V_m)$.*

Adaptive Doubly Stochastic Gradient Algorithm

In this section, we follow the the relationship between statistical accuracy and regularized empirical risk $R_n(w)$ revealed in Remark 2 to propose our adaptive doubly stochastic gradient (*i.e.*, AdaDSG) algorithm. Next, we also provide the complexity analysis of AdaDSG algorithm.

AdaDSG Algorithm

As mentioned in Remark 2, we consider two subsets $\mathcal{S}_m$ and $\mathcal{S}_{2m}$ of the full i.i.d. training set such that $\mathcal{S}_m \subset \mathcal{S}_{2m}$. The conclusion suggests that there is no benefit to solve the subproblem R_m beyond its statistical accuracy. Thus, we start by a small number of samples and use an inner solver to solve the corresponding problem with its statistical accuracy. After that, we double the size of the training set and use the solution of the previous problem with half samples as a warm start for the new problem. We repeat this procedure until the selected training set becomes identical to the given training set $\mathcal{S}$ which contains n samples. We summarize our AdaDSG algorithm in Algorithm 1.

In this paper, we use the vanilla doubly stochastic gradient descent (DSGD) algorithm to solve multiple subproblems $R_m(w)$ which is summarized in Algorithm 2. Specifically, we randomly select a pair of samples (x_i, y_i) and (x_j, y_j) at the t-th iteration, and compute the stochastic gradient on the pair of samples (x_i, y_i) and (x_j, y_j) as follows:

$$\nabla F_{i,j}(w^t) = \nabla L(f_{w^t}, (x_i, y_i), (x_j, y_j)) + cV_m w^t \tag{8}$$

Given the learning rate $\gamma^t = \frac{1}{\mu(t+1)}$ where μ is the strong convexity parameter defined in Assumption 2, we update the solution as $w^{t+1} \leftarrow w^t - \gamma^t \nabla F_{i,j}(w^t)$.

Remark 3 (Difference to the existing algorithms)**.** *AdaDSG is different to the existing adaptive sample size algorithms (Mokhtari et al., 2016; Mokhtari and Ribeiro, 2017) in checking the termination condition of the inner solver. Theorem 4 suggests that running DSGD only with $\mathcal{O}(m)$ steps can reach the statistical accuracy V_m for the subproblem R_m. Our AdaDSG runs a fixed number (*i.e.*, $O(m)$) of iterations instead of explicitly checking the termination condition $\|\nabla R_m(w_m)\| < \sqrt{2c}V_m$ as did in the existing adaptive sample size algorithms.*

Algorithm 1 Adaptive doubly stochastic gradient algorithm (AdaDSG)

```
Input: Initial sample size $m_0$, and initial solution $w^0$ such
    that $R_{m_0}(w^0) - R_{m_0}(w^*_{m_0}) \leq V_{m_0}$.
Output: $w^S$.
Initialize $w^s = w^0$ and $m = m_0$.
while $m \leq n$ do
    Increase the samples sizes $m = \min\{2 \times m, n\}$.
    Call DSGD to solve $R_m(w)$ with the initial solution
        $w^s$ and an inner loop number $\mathcal{O}(m)$.
    Set $w^{s+1} = \tilde{w}$, where $\tilde{w}$ is the solution returned by
        DSGD on the training set with the size of $m$.
end while
```

Algorithm 2 DSGD algorithm

```
Input: Learning rate $\gamma^t = \frac{1}{\mu(t+1)}$, loop number $T$, and ini-
    tial solution $w^0$.
Output: $w^T$.
for $t = 0, 1, 2, \cdots, T-1$ do
    Pick $(x_i, y_i)$ and $(x_j, y_j)$ uniformly at random from
        the set $\mathcal{S}_m$.
    Update $w^{t+1} \leftarrow w^t - \gamma^t \nabla F_{i,j}(w^t)$.
end for
```

Complexity Analysis

We first give the assumptions of Lipschitz smoothness (Assumption 1), strong convexity (Assumption 2) and bounded variance (Assumption 3), which are critical to the analysis of our AdaDSG.

Assumption 1 (Lipschitz smoothness)**.** *The function $F_{i,j}$ ($\forall i \in \mathcal{S}$ and $\forall j \in \mathcal{S}$) in (3) is Lipschitz smooth with the Lipschitz constant $L \geq cV_n$, which means that $\forall w \in \mathbb{R}^d$ and $\forall w' \in \mathbb{R}^d$, we have:*

$$\|\nabla F_{i,j}(w) - \nabla F_{i,j}(w')\| \leq L\|w - w'\| \tag{9}$$

As shown in the formulation (3), $F_{i,j}$ includes a regularization term $\frac{cV_n}{2}\|w\|^2$. If the pairwise loss function L is smooth, we have that $F_{i,j}$ is at least cV_n-Lipschitz smooth.

Assumption 2 (Strong convexity)**.** *The differentiable function $R_n(w)$ in (3) is strongly convex with parameter $\mu \geq cV_n$, which means that $\forall w \in \mathbb{R}^d$ and $\forall w' \in \mathbb{R}^d$, we have*

$$R_n(w) \geq R_n(w') + \langle \nabla R_n(w'), w - w' \rangle + \frac{\mu}{2}\|w - w'\|^2 \tag{10}$$

If the pairwise loss function L is convex, we have that $R_n(w)$ is at least cV_n-strongly convex.

Assumption 3 (Bounded variance)**.** *We assume that the second moment of the stochastic gradient generated from DSGD algorithm is upper bounded. Specifically, given an initial solution w^0, there exists a constant $\bar{c}$ such that*

$$\mathbb{E}\|\nabla F_{i,j}(w^t)\|^2 \leq \bar{c}\left(R_n(w^0) - R_n(w^*_n)\right) \tag{11}$$

Based on Assumptions 1, 2 and 3, we prove the following conclusions. The detailed proof can be found in our Appendix.

1. The inner loop number for DSGD is $\mathcal{O}(m')$ which can guarantee $\mathbb{E}\left[R_{m'}(w_{m'}) - R_{m'}(w^*_{m'})\right] \leq V_{m'}$ (*i.e.*, Theorem 4).
2. The overall computational complexity of AdaDSG is $\mathcal{O}(n)$ which can guarantee AdaDSG to achieve the statistical accuracy on the full training set (*i.e.*, Theorem 5).

Before proving Theorem 4, we provide Lemma 1 which shows that DSGD algorithm has a sublinear convergence rate.

Lemma 1. *Suppose Assumptions 1, 2 and 3 hold. For the DSGD Algorithm, we have*

$$\begin{aligned} & \mathbb{E}R_n(w^T) - R_n(w^*) \qquad (12) \\ \leq \; & \frac{L\max\{\|w^0 - w^*\|^2, \frac{\bar{c}}{\mu^2}\left(R_n(w^0) - R_n(w^*)\right)\}}{2T} \end{aligned}$$

Remark 4. *Lemma 1 provides a sublinear linear convergence rate to DSGD algorithm which is similar to the one of traditional SGD algorithm. Further, according to Lemma 1, we have that the overall computational complexity of DSGD is $\mathcal{O}(n\sqrt{n})$ [1] to make the solution satisfy $\mathbb{E}\left[R_n(w_n) - R_n(w^*_n)\right] \leq V_n$. To highly reduce the overall computational cost of achieving the statistical accuracy on the whole samples, we propose an adaptive sample size version to DSGD (i.e., AdaDSG).*

Theorem 4. *Consider the variable w_m as a V_m-suboptimal solution of the risk R_m in expectation,* i.e., *$\mathbb{E}\left[R_m(w_m) - R_m(w^*_m)\right] \leq V_m$. Consider the sets $\mathcal{S}_m \subset \mathcal{S}_{m'} \subseteq \mathcal{S}$ such that $m' = 2m$, and suppose Assumptions 1, 2 and 3 hold. To make the solution of DSGD satisfy $\mathbb{E}\left[R_{m'}(w_{m'}) - R_{m'}(w^*_{m'})\right] \leq V_{m'}$, the inner loop number $T_{m'}$ of DSGD at the stage of $\mathcal{S}_{m'}$ should satisfy:*

$$T_{m'} \geq \frac{\max\left\{2\sqrt{m'}, \frac{\bar{c}m'}{c}\right\} L\left(5 + (1 - \frac{1}{\sqrt{2}})\frac{c}{2}\|w^*\|^2\right)}{\sqrt{2}c} \tag{13}$$

[1] This conclusion can be easily derived from Lemma 1.

Remark 5. *Let w_m (a V_m-suboptimal solution of R_m) be the initial solution of the problem $R_{m'}$. Theorem 4 clearly shows that, if we want to have $\mathbb{E}\left[R_{m'}(w_{m'}) - R_{m'}(w^*_{m'})\right] \leq V_{m'}$, we only need to run the DSGD algorithm with $\mathcal{O}(m')$ inner loops.*

Based on Theorem 4, we provide the complexity analysis of AdaDSG in Theorem 5.

Theorem 5. *Suppose Assumptions 1, 2 and 3 hold. To reach the statistical accuracy V_n on the full training set $\mathcal{S}$, the overall computational complexity of AdaDSG is given by*

$$\max\left\{\sqrt{n}\frac{2\sqrt{2}}{\sqrt{2}-1}, \frac{2\bar{c}n}{c}\right\} \frac{L\left(5+(1-\frac{1}{\sqrt{2}})\frac{c}{2}\|w^*\|^2\right)}{c\sqrt{2}} \quad (14)$$

Remark 6. *Theorem 5 shows that, the overall computational complexity of AdaDSG is $\mathcal{O}(n)$ to make the solution satisfy the statistical accuracy (*i.e.*, $\mathbb{E}\left[R_n(w_n) - R_n(w^*_n)\right] \leq V_n$). Compared with the overall computational complexity $\mathcal{O}(n\sqrt{n})$ of DSGD, our AdaDSG algorithm is much more efficient. To the best of our knowledge, the overall computational complexity $\mathcal{O}(n)$ is the best theoretical result for pairwise learning to achieve the statistical accuracy.*

Experimental Results

Experimental Setup

Design of Experiments: Because there has been great interest in AUC maximization in recent data science research, we consider the pairwise learning on the AUC maximization problem in this paper. We conduct experiments not only to verify the effectiveness of our AdaDSG algorithm, but also to show that our AdaDSG algorithm has significantly better scalability and efficiency than the existing pairwise learning algorithms.

To verify the effectiveness of AdaDSG, we compare the convergence speeds of DSGD and AdaDSG by observing the AUC on the testing set vs. iteration number curves. To verify the superiority of our AdaDSG algorithm on the scalability and efficiency, we compare the AUC on the testing set vs. training time for different AUC maximization algorithms. The state-of-the-art AUC maximization algorithms compared in the experiments are the online pairwise (OLP) algorithm (Kar et al., 2013), the OPAUC algorithm (Gao et al., 2013), the SOLAM algorithm (Ying, Wen, and Lyu, 2016) and our AdaDSG algorithms which are summarized in Table 1.

Implementation Details: Our experiments were performed on an 8-core Intel Xeon E3-1240 machine. We implemented our AdaDSG algorithm in MATLAB. We used the MATLAB code from `http://lamda.nju.edu.cn/files/OPAUC.zip` as the implementation of the OPAUC algorithm. We used the MATLAB code from `https://www.albany.edu/~yy298919/nips16_solam.zip` as the implementation of the SOLAM algorithm. We used the MATLAB and C mixed code from `https://www.cse.iitk.ac.in/users/purushot/code.php` as the implementation of the OLP algorithm (Kar et al., 2013), where the core function was implemented by C. Note that, even though C implementation is significantly more efficient than a pure MATLAB implementation, the experimental results still show that our AdaDSG with MATLAB code is much faster than OLP with C code.

Table 3: The real-world dasetsets used in the experiments.

Dataset	Feature size	Sample size
A9a	123	32,561
Covtype	54	581,012
Ijcnn1	22	49,990
Phishing	68	11,055
Usps	256	7,291
Mnist	780	60,000
Rcv1	47,236	20,242
Real-sim	20,958	72,309

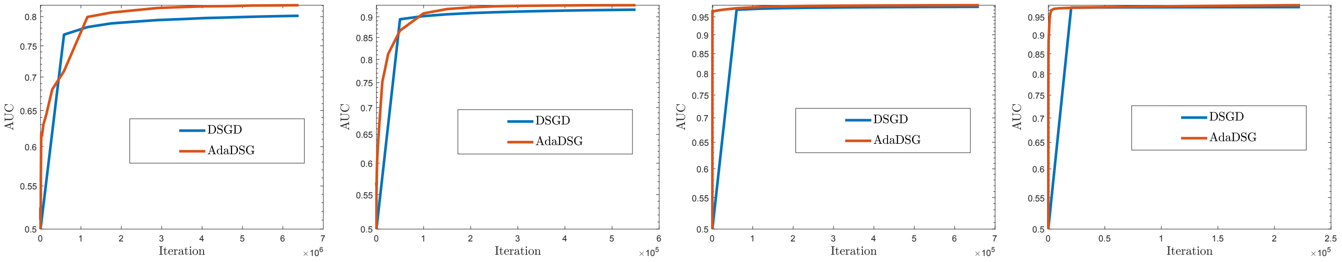
For the OLP algorithm, we set the parameter $s = 500$ in the experiments. For the OPAUC algorithm on high dimensional datasets (feature size larger than 10,000), we used the low-rank version, and set the rank parameter $\tau = 100$. For our AdaDSG algorithm, the initial learning rate γ^0 was tuned from 1 to 10^{-4}, and the outer loop number was set as 20. In each experiment, the AUC value is the average of 25 trials. We randomly partitioned each dataset into 75% for training and 25% for testing. Regularization parameters were used in (Gao et al., 2013) and our model. We fixed the regularization parameters as 1 in our experiments. In the implementation of our AdaDSG algorithm, we set $V_n = \frac{1}{\sqrt{n}}$, and set the inner loop number of DSGD for the subproblem R_m as m.

Datasets: Table 3 summarizes the eight real-world benchmark datasets used in our experiments. They are the A9a, Covtype, Ijcnn1, Phishing, Usps, Mnist, Rcv1 and Real-sim datasets from the LIBSVM repository[2]. For multi-class datasets (*i.e.*, Usps and Mnist), we transformed them into binary classification problems by randomly partitioning the data into two groups, where each group includes the same number of classes. Please note that, to test the scalability of different algorithms, all the datasets used in the experiments are with large sample size or large feature size.

Results and Discussions

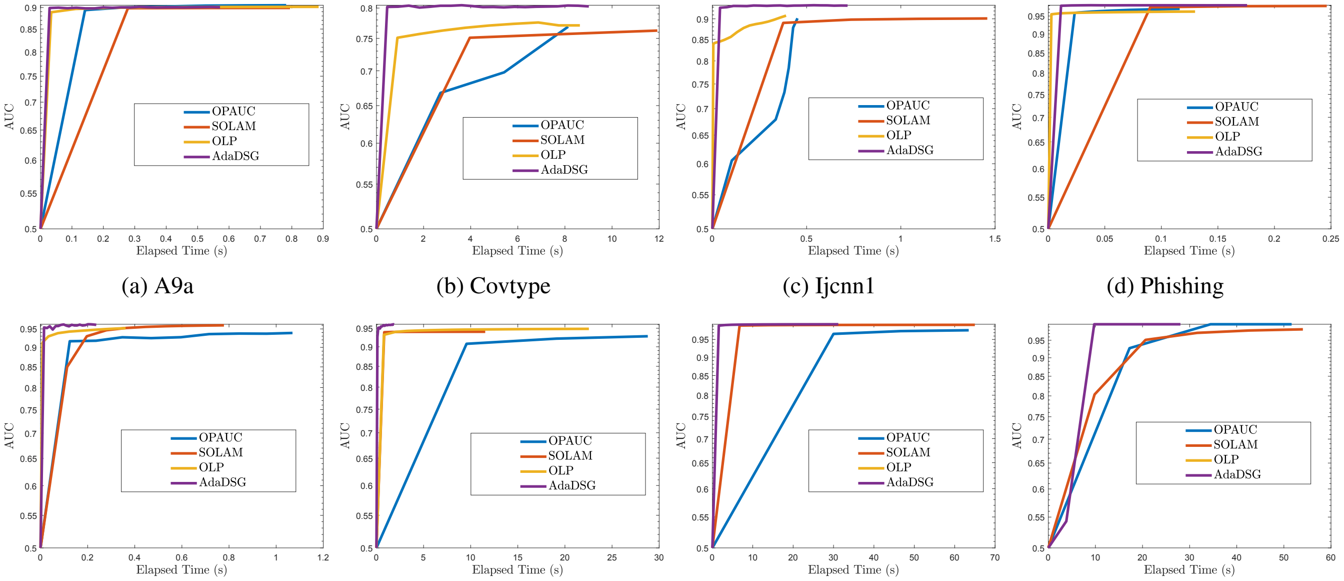
Figure 1 provides the convergence results of testing AUC vs. iteration number of our AdaDSG algorithm and DSGD algorithm on the Covtype, Ijcnn1, Mnist and Rcv1 datasets. The results show that AdaDSG can converge to a good AUC value with less time compared with DSGD. The results verify the effectiveness of AdaDSG, *i.e.*, AdaDSG reduces a lot of computing time to achieve the statistical accuracy on the whole samples, which supports the theoretical result in Remark 6.

[2] The LIBSVM repository is available at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.



(a) Covtype (b) Ijcnn1 (c) Mnist (d) Rcv1

Figure 1: Testing AUC vs. iteration number curves of our AdaDSG algorithm and DSGD algorithm.



(a) A9a (b) Covtype (c) Ijcnn1 (d) Phishing

(e) Usps (f) Mnist (g) Rcv1 (h) Real-sim

Figure 2: Testing AUC vs. training time curves of AdaDSG algorithm and three state-of-the-art AUC maximization algorithms, *i.e.*, the OLP (Kar et al., 2013), OPAUC (Gao et al., 2013) and SOLAM (Ying, Wen, and Lyu, 2016) algorithms. Note that the OLP curves are missing on the Rcv1 and Real-sim datasets because the program of OLP crashes on these large-scale datasets.

Figure 2 provides the convergence results of testing AUC vs. training time of AdaDSG algorithm and three state-of-the-art AUC maximization algorithms, *i.e.*, the OLP (Kar et al., 2013), OPAUC (Gao et al., 2013), and SOLAM (Ying, Wen, and Lyu, 2016) algorithms on the A9a, Covtype, Ijcnn1, Phishing, Usps, Mnist, Rcv1 and Real-sim datasets. Please note that the OLP curves are missing on the Rcv1 and Real-sim datasets because the program of OLP crashes on these large-scale datasets. Although the OLP (Kar et al., 2013) and SOLAM (Ying, Wen, and Lyu, 2016) algorithms solve the empirical risk (2) and the OPAUC (Gao et al., 2013) and our AdaDSG algorithms solve the regularized empirical risk (3), the results still clearly show that our AdaDSG has significantly better scalability and efficiency than the existing pairwise learning algorithms.

Conclusion

In this paper, we first analyzed the relationship between the statistical accuracy and the regularized empire risk for pairwise loss. Based on the relationship, we proposed a scalable and efficient adaptive doubly stochastic gradient algorithm (*i.e.*, AdaDSG) for regularized pairwise learning problems. We believe AdaDSG is a breakthrough to pairwise learning for the following four reasons. First, AdaDSG works for general forms of pairwise learning problems, loss functions and hypothesis functions. Second, the pivotal step of AdaDSG is computing doubly stochastic gradients on a pair of samples which make the computation of AdaDSG much scalable and efficient. Third and most importantly, we prove that the overall computational cost of AdaDSG is $\mathcal{O}(n)$ to reach the statistical accuracy $\mathcal{O}(\frac{1}{\sqrt{n}})$ on the training set with the size of n, which is the best theoretical result for pairwise learning to the best of our knowledge. At last, we conducted the experiments on the application of the AUC maximization. The experimental results on real-world benchmark datasets not only confirm the effectiveness of AdaDSG, but also show that AdaDSG has significantly better scalability and efficiency than existing pairwise learning algorithms.

References

Agarwal, S., and Niyogi, P. 2009. Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research* 10(Feb):441–474.

Agarwal, S.; Graepel, T.; Herbrich, R.; Har-Peled, S.; and Roth, D. 2005. Generalization bounds for the area under the roc curve. *Journal of Machine Learning Research* 6(Apr):393–425.

Boissier, M.; Lyu, S.; Ying, Y.; and Zhou, D.-X. 2016. Fast convergence of online pairwise learning algorithms. In *Artificial Intelligence and Statistics*, 204–212.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Bradley, J. K.; Kyrola, A.; Bickson, D.; and Guestrin, C. 2011. Parallel coordinate descent for l 1-regularized loss minimization. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 321–328. Omnipress.

Cao, Q.; Guo, Z.-C.; and Ying, Y. 2016. Generalization bounds for metric and similarity learning. *Machine Learning* 102(1):115–132.

Cesa-Bianchi, N.; Conconi, A.; and Gentile, C. 2004. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory* 50(9):2050–2057.

Dai, B.; Xie, B.; He, N.; Liang, Y.; Raj, A.; Balcan, M.-F. F.; and Song, L. 2014. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems*, 3041–3049.

Daneshmand, H.; Lucchi, A.; and Hofmann, T. 2016. Starting small-learning with adaptive sample sizes. In *International conference on machine learning*, 1463–1471.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, 1646–1654.

Eisen, M.; Mokhtari, A.; and Ribeiro, A. 2018. Large scale empirical risk minimization via truncated adaptive newton method. In *International Conference on Artificial Intelligence and Statistics*, 1447–1455.

Gao, W.; Jin, R.; Zhu, S.; and Zhou, Z.-H. 2013. One-pass auc optimization. In *International Conference on Machine Learning*, 906–914.

Gu, B.; Huo, Z.; Deng, C.; and Huang, H. 2018. Faster derivative-free stochastic algorithm for shared memory machines. In *International Conference on Machine Learning*, 1807–1816.

Gu, B.; Huo, Z.; and Huang, H. 2018. Asynchronous doubly stochastic group regularized learning. In *International Conference on Artificial Intelligence and Statistics*, 1791–1800.

Jin, R.; Wang, S.; and Zhou, Y. 2009. Regularized distance metric learning: Theory and algorithm. In *Advances in neural information processing systems*, 862–870.

Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 315–323.

Kar, P.; Sriperumbudur, B.; Jain, P.; and Karnick, H. 2013. On the generalization ability of online learning algorithms for pairwise loss functions. In *International Conference on Machine Learning*, 441–449.

Kumar, A.; Niculescu-Mizil, A.; Kavukcoglu, K.; and Daumé, H. 2012. A binary classification framework for two-stage multiple kernel learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 1331–1338. Omnipress.

Lei, Y.; Lin, S.-B.; and Tang, K. 2018. Generalization bounds for regularized pairwise learning. In *IJCAI*, 2376–2382.

Lin, J.; Lei, Y.; Zhang, B.; and Zhou, D.-X. 2017. Online pairwise learning algorithms with convex loss functions. *Information Sciences* 406:57–70.

Liu, J., and Wright, S. J. 2015. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization* 25(1):351–376.

Mokhtari, A., and Ribeiro, A. 2017. First-order adaptive sample size methods to reduce complexity of empirical risk minimization. In *Advances in Neural Information Processing Systems*, 2057–2065.

Mokhtari, A.; Daneshmand, H.; Lucchi, A.; Hofmann, T.; and Ribeiro, A. 2016. Adaptive newton method for empirical risk minimization to statistical accuracy. In *Advances in Neural Information Processing Systems*, 4062–4070.

Rejchel, W. 2012. On ranking and generalization bounds. *Journal of Machine Learning Research* 13(May):1373–1392.

Weinberger, K. Q., and Saul, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10(Feb):207–244.

Ying, Y., and Li, P. 2012. Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research* 13(Jan):1–26.

Ying, Y.; Wen, L.; and Lyu, S. 2016. Stochastic online auc maximization. In *Advances in Neural Information Processing Systems*, 451–459.

Yu, N. 2013. Gradient methods for minimizing composite objective function. *Math. Program* 140(1):125–161.

Zhao, P.; Hoi, S. C.; Jin, R.; and Yang, T. 2011. Online auc maximization. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 233–240. Omnipress.

Zhao, T.; Yu, M.; Wang, Y.; Arora, R.; and Liu, H. 2014. Accelerated mini-batch randomized block coordinate descent method. In *Advances in neural information processing systems*, 3329–3337.