Constrained Stochastic Gradient Descent for Large-scale Least Squares Problem

Yang Mu
University of Massachusetts
Boston
100 Morrissey Boulevard
Boston, MA, US 02125
yangmu@cs.umb.edu

Wei Ding[*]
University of Massachusetts Boston
100 Morrissey Boulevard
Boston, MA, US 02125
ding@cs.umb.edu

Tianyi Zhou
University of Technology Sydney
235 Jones Street
Ultimo, NSW 2007, Australia
tianyi.david.zhou@gmail.com

Dacheng Tao
University of Technology Sydney
235 Jones Street
Ultimo, NSW 2007, Australia
dacheng.tao@uts.edu.au

ABSTRACT

The least squares problem is one of the most important regression problems in statistics, machine learning and data mining. In this paper, we present the Constrained Stochastic Gradient Descent (CSGD) algorithm to solve the large-scale least squares problem. CSGD improves the Stochastic Gradient Descent (SGD) by imposing a provable constraint that the linear regression line passes through the mean point of all the data points. It results in the best regret bound $O(\log T)$, and fastest convergence speed among all first order approaches. Empirical studies justify the effectiveness of CSGD by comparing it with SGD and other state-of-the-art approaches. An example is also given to show how to use CSGD to optimize SGD based least squares problems to achieve a better performance.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Least squares methods, Stochastic programming; I.2.6 [**Learning**]: Parameter learning

General Terms

Algorithms, Theory

Keywords

Stochastic optimization, Large-scale least squares, online learning

1. INTRODUCTION

The stochastic least squares problem aims to find the coefficient $\mathbf{w} \in \mathbb{R}^d$ to minimize the objective function at step t

$$\mathbf{w}_{t+1}^* = \arg\min_{\mathbf{w}} \frac{1}{t} \sum_{i=1}^{t} \frac{1}{2} \left\| y_i - \mathbf{w}^T \mathbf{x}_i \right\|_2^2, \tag{1}$$

where $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ is an input-output pair randomly drawn from data set $(\mathbf{X}, \mathbf{Y})$ endowed in a distribution $\mathcal{D}$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, $\mathbf{w}_{t+1}^* \in \mathbb{R}^d$ is a parameter that minimizes the empirical least squares loss at step t, and $l(\mathbf{w}, \mathbf{x}_t, y_t) = \frac{1}{2} \left\| y_t - \mathbf{w}^T \mathbf{x}_t \right\|_2^2$ is the empirical risk.

For large-scale problems, the classical optimization methods, such as interior point method and conjugate gradient descent, have to scan all data points several times in order to evaluate the objective function and find the optimal $\mathbf{w}^*$. Recently, Stochastic Gradient Descent (SGD) [7, 29, 9, 3, 20, 13] methods show its promising efficiency in solving large-scale problems. Some of them have been widely applied to the least squares problem. The Least Mean Squares (LMS) algorithm [25] is the standard first order SGD, which takes a scalar as the learning rate. The Recursive Least Squares (RLS) approach [25, 15] is an instantiation of the stochastic Newton method by replacing the scalar learning rate with an approximation of the Hessian matrix inverse. The Averaged Stochastic Gradient Descent (ASGD) [21] averages the SGD results to estimate $\mathbf{w}^*$. ASGD converges more stably than SGD. Its convergence rate even approaches to that of second order method, when the estimator is sufficiently close to $\mathbf{w}^*$. This happens after processing a huge amount of data points.

Since the least squares loss $l(\mathbf{w}_t, \mathbf{X}, \mathbf{Y})$ is usually strongly convex [5], the first order approaches can converge at the rate of $O(1/T)$. Given the smallest eigenvalue λ_0 of the Hessian matrix [8, 7, 18], many algorithms can achieve fast convergence rates and good regret bounds [7, 2, 10, 23, 11]. However, if the Hessian matrix is unknown in advance, SGD may perform poorly [18].

For high-dimensional large-scale problems, the strong convexity are not always guaranteed, because the smallest eigenvalue of the Hessian matrix might be close to 0. Without

[*]Corresponding author

KDDqŕ13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 20XX ACM 978-1-4503-2174-7/13/08 ...$15.00.

the assumption of the strong convexity, the convergence rate of the first order approaches reduces to $O(1/\sqrt{T})$ [30] while retains computation complexity of $O(d)$ in each iteration. Second order approaches, using the Hessian approximation, converge at rate $O(1/T)$. Although they are appealing due to the fast convergence rate and stableness, the expensive time complexity of $O(d^2)$ when dealing with each iteration limits the use of second order approaches in practical large-scale problems.

In this paper, we prove that the linear regression line defined by the optimal coefficient $\mathbf{w}^*$ passes through the mean point $(\bar{\mathbf{x}}, \bar{y})$ of all the data points drawn from the distribution $\mathcal{D}$. Given this property, we can significantly improve SGD for optimizing the large-scale least squares problem by adding an equality constraint $\mathbf{w}_{t+1}^T\bar{\mathbf{x}}_\mathbf{t} - \bar{y}_t = 0$, where $(\bar{\mathbf{x}}_\mathbf{t}, \bar{y}_t)$ is the batch mean of the collected data points till step t during the optimization iterations. The batch mean $(\bar{\mathbf{x}}_\mathbf{t}, \bar{y}_t)$ is an unbiased estimation to $(\bar{\mathbf{x}}, \bar{y})$ by iterations (*cf*. the Law of Large Numbers). We term the proposed approach as the constrained SGD (CSGD). CSGD shrinks the optimal solution space of the least squares problem from the entire $\mathbb{R}^d$ to a hyper-plane in $\mathbb{R}^d$, thus significantly improves the convergence rate and the regret bound. In particular,

- Without the strong convexity assumption, CSGD converges at the rate of $O(\log T/T)$, which is close to that of a full second order approach, while retaining time complexity of $O(d)$ in each iteration.
- CSGD achieves the $O(\log T)$ regret bound without requiring strong convexity, which is the best regret bound among existing SGD methods.

Note that when the data points are centralized (mean is $\mathbf{0}$), the constraint becomes trivial and CSGD reduces to SGD, which is the worst case for CSGD. In practical online learning, the collected data points, however, are often not centralized, and thus CSGD is preferred. In this paper, we only discuss the properties of CSGD when data points are not centralized.

Notations. We denote the input data point $\mathbf{x}_t = [1\ \tilde{\mathbf{x}}_t]^T = [x_t^{(1)}, \ldots, x_t^{(d)}]^T$ and $\mathbf{w}_t = [w_t^{(1)}\ \tilde{\mathbf{w}}_t]^T = [w_t^{(1)}, \ldots, w_t^{(d)}]^T$, where $x_t^{(1)} = 1$ is the first element of $\mathbf{x}_t$ and $w_t^{(1)}$ is the bias parameter [6]. $\|\cdot\|_p$ is the L_p norm, $\|\cdot\|_p^2$ is the squared L_p norm, $|\cdot|$ is the absolute operation for scalars and $l(\mathbf{w})$ is abbreviated for $l(\mathbf{w}, \mathbf{x}_t, y_t)$.

2. CONSTRAINED STOCHASTIC GRADIENT DESCENT

We present the Constrained Stochastic Gradient Descent (CSGD) algorithm for the large-scale least squares problem by incorporating SGD with the fact that the linear regression line passes through the mean point of all the data points.

2.1 CSGD algorithm

The standard Stochastic Gradient Descent (SGD) algorithm takes the form of

$$\mathbf{w}_{t+1} = \Pi_\tau(\mathbf{w}_t - \eta_t \mathbf{g}_t), \tag{2}$$

where η_t is an appropriate learning rate, $\mathbf{g}_t$ is the gradient of the loss function $l(\mathbf{w}, \mathbf{x}_t, y_t) = \frac{1}{2}\left\|y_t - \mathbf{w}^T\mathbf{x}_t\right\|_2^2$, $\Pi_\tau(\cdot)$ is the Euclidean projection function that projects $\mathbf{w}$ onto the predefined convex set τ by

$$\Pi_\tau(\mathbf{w}) = \arg\min_{\mathbf{v}\in\tau} \|\mathbf{v} - \mathbf{w}\|_2^2 . \tag{3}$$

In least squares, $\mathbf{w}$ is defined in the entire $\mathbb{R}^d$, and $\Pi_\tau(\cdot)$ can be taken off. Thus, the search space of SGD is the entire $\mathbb{R}^d$ to obtain the optimal solution.

According to Theorem 2.1 (*cf*. Section 2.2), we add a constraint $\mathbf{w}^T\bar{\mathbf{x}}_t - \bar{y}_t = 0$ at step t to SGD, where $(\bar{\mathbf{x}}_t, \bar{y}_t)$ is an unbiased estimation to $(\bar{\mathbf{x}}, \bar{y})$ after t iterations, and obtain CSGD

$$\mathbf{w}_{t+1}^* = \arg\min_{\mathbf{w}} \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\left\|y_i - \mathbf{w}^T\mathbf{x}_i\right\|_2^2, \text{s.t. } \mathbf{w}^T\bar{\mathbf{x}}_t - \bar{y}_t = 0. \tag{4}$$

The constraint in Eq.(4) determines the hyper-plane $\tau_t = \{\mathbf{w}|\mathbf{w}^T\bar{\mathbf{x}}_t = \bar{y}_t\}$ residing in $\mathbb{R}^d$.

By replacing τ with τ_t in Eq.(2), we have

$$\mathbf{w}_{t+1} = \Pi_{\tau_t}(\mathbf{w}_t - \eta_t\mathbf{g}_t). \tag{5}$$

The projection function $\Pi_{\tau_t}(\cdot)$ projects a point onto the hyper-plane τ_t. By solving Eq.(3), $\Pi_{\tau_t}(\cdot)$ is uniquely defined at each step by

$$\Pi_{\tau_t}(\mathbf{v}) = \mathbf{P}_t\mathbf{v} + \mathbf{r}_t, \tag{6}$$

where $\mathbf{P}_t$ is the projection matrix at step t and takes the form of

$$\mathbf{P}_t = \mathbf{I} - \frac{\bar{\mathbf{x}}_t\bar{\mathbf{x}}_t^T}{\|\bar{\mathbf{x}}_t\|_2^2}, \tag{7}$$

where $\mathbf{P}_t \in \mathbb{R}^{d\times d}$ is idempotent and projects a vector onto the subspace generated by $\mathbf{x}_t$, and $\mathbf{r}_t = \frac{\bar{y}_t}{\|\bar{\mathbf{x}}_t\|_2^2}\bar{\mathbf{x}}_t$.

By combining Eqs.(5) and (6), the iterative procedure for CSGD is

$$\mathbf{w}_{t+1} = \mathbf{P}_t(\mathbf{w}_t - \eta_t\mathbf{g}_t) + \mathbf{r}_t. \tag{8}$$

We can obtain the time and space complexities of above procedure both as $O(d)$ after plugging Eq.(7) into (8) and update $\mathbf{w}_{t+1}$,

$$\begin{aligned}\mathbf{w}_{t+1} &= \left(\mathbf{I} - \frac{\bar{\mathbf{x}}_t\bar{\mathbf{x}}_t^T}{\|\bar{\mathbf{x}}_t\|_2^2}\right)(\mathbf{w}_t - \eta_t\mathbf{g}_t) + \mathbf{r}_t \\ &= \mathbf{w}_t - \eta_t\mathbf{g}_t - \bar{\mathbf{x}}_t\left(\bar{\mathbf{x}}_t^T(\mathbf{w}_t - \eta_t\mathbf{g}_t)\right)/\|\bar{\mathbf{x}}_t\|_2^2 + \mathbf{r}_t.\end{aligned} \tag{9}$$

Algorithm 1 describes how to calculate CSGD for the least squares problem. This algorithm has the time and space complexities both of $O(d)$.

Algorithm 1 Constrained Stochastic Gradient Descent (CSGD)

```
Initialize $\mathbf{w}_1 = \bar{\mathbf{x}}_0 = \mathbf{0}$ and $\bar{y}_0 = 0$.
for $t = 1, 2, 3, \ldots$ do
    Compute the gradient $\mathbf{g}_t \in \partial l(\mathbf{w}_t, \mathbf{x}_t, y_t)$.
    Compute $(\bar{\mathbf{x}}_t, \bar{y}_t)$ with
    $\bar{\mathbf{x}}_t = \frac{t-1}{t}\bar{\mathbf{x}}_{t-1} + \frac{1}{t}\mathbf{x}_t$, and
    $\bar{y}_t = \frac{t-1}{t}\bar{y}_{t-1} + \frac{1}{t}y_t$.
    Compute
    $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\mathbf{g}_t - \bar{\mathbf{x}}_t\left(\bar{\mathbf{x}}_t^T(\mathbf{w}_t - \eta_t\mathbf{g}_t)\right)/\|\bar{\mathbf{x}}_t\|_2^2 + \mathbf{r}_t$.
end for
```

2.2 Regression line constraint

Algorithm 1 relies on the fact that the optimal solution lies in a hyper-plane decided by the mean point, which leads to a significant improvement on the convergence rate and the regret bound.

THEOREM 2.1. (Regression line constraint) *The optimal solution $\mathbf{w}^*$ lies on the hyper-plane, $\mathbf{w}^T\bar{\mathbf{x}} - \bar{y} = 0$, which is defined by the mean point $(\bar{\mathbf{x}}, \bar{y})$ of data points drawn from the distribution $\mathcal{X} \times \mathcal{Y}$ endowed in $\mathcal{D}$.*

PROOF. The loss function is explicitly defined as

$$l(\mathbf{w}^*, \mathcal{X}, \mathcal{Y}) = \sum_{(x_t, y_t) \in \mathcal{D}} \frac{1}{2} \|y_t - \tilde{\mathbf{w}}^{*T} \tilde{\mathbf{x}}_t - \mathbf{w}^{*(1)}\|_2^2, \qquad (10)$$

where $\mathbf{w}^{*(1)}$ is the first element of $\mathbf{w}^*$.

Setting the derivative of the loss function w.r.t. $\mathbf{w}^{*(1)}$ to zero, we obtain

$$\sum_{\mathbf{x}_t \in \mathcal{X}} \tilde{\mathbf{w}}^{*T} \tilde{\mathbf{x}}_t + \mathbf{w}^{*(1)} - \sum_{y_t \in \mathcal{Y}} y_t = 0.$$

$$\sum_{\mathbf{x}_t \in \mathcal{X}} \mathbf{w}^{*T} \mathbf{x}_t - \sum_{y_t \in \mathcal{Y}} y_t = 0.$$

Thus the optimal solution $\mathbf{w}^*$ satisfies $\mathbf{w}^T\bar{\mathbf{x}} - \bar{y} = 0$. □

Theorem 2.1 is the core theorem for our method. Bishop [6] applied the derivitive w.r.t the bias $\mathbf{w}^{*(1)}$ to study the property of the bias. However, although the theorem itself is in a simple form, to the best of our knowledge, it has never been stated and applied in any approach for least squares optimization.

The mean point $(\bar{\mathbf{x}}, \bar{y})$ over a distribution $\mathcal{D}$ is usually not given. In a stochastic approach, we can use the batch mean $(\bar{\mathbf{x}}_t, \bar{y}_t)$ to approximate $(\bar{\mathbf{x}}, \bar{y})$. The approximation has an estimation error, however, it will not lower the performance. This is because the batch optimal always satisfies this constraint when optimizing the empirical loss.

Therefore, we give the constrained estimation error bound for completeness.

PROPOSITION 2.2. (Constrained estimation error bound) *According to the Law of Large Numbers, we assume there is a step $m \leq T$ yeilds $\|\mathbf{w}^*\|_2 \|\bar{\mathbf{x}}_m - \bar{\mathbf{x}}\|_2 + |\bar{y}_m - \bar{y}| \leq \epsilon \|\mathbf{w}^*\|_2$. Then given a tolerable small value ϵ, the estimation error bound $\|\Pi_{\tau_t}(\mathbf{w}^*) - \mathbf{w}^*\|_2 \leq \epsilon$ holds for any step $t \geq m$.*

PROOF. Since $\|\Pi_{\tau_t}(\mathbf{w}^*) - \mathbf{w}^*\|_2$ is the distance between $\mathbf{w}^*$ and the hyper-plane τ_t, we have

$$\|\Pi_{\tau_t}(\mathbf{w}^*) - \mathbf{w}^*\|_2^2 = \frac{|\bar{y}_t - \mathbf{w}^{*T} \bar{\mathbf{x}}_t|}{\|\mathbf{w}^*\|_2}.$$

Along with $\mathbf{w}^{*T}\bar{\mathbf{x}} - \bar{y} = 0$, we have

$$\begin{aligned} &|\bar{y}_t - \mathbf{w}^{*T} \bar{\mathbf{x}}_t| \qquad (11) \\ =&\|\bar{y}_t - \bar{y} - (\mathbf{w}^{*T}\bar{\mathbf{x}}_t - \mathbf{w}^{*T}\bar{\mathbf{x}})\|_2 \\ \leq&\|\mathbf{w}^*\|_2 \|\bar{\mathbf{x}}_t - \bar{\mathbf{x}}\|_2 + |\bar{y}_t - \bar{y}| \\ \leq&\|\mathbf{w}^*\|_2 \|\bar{\mathbf{x}}_m - \bar{\mathbf{x}}\|_2 + |\bar{y}_m - \bar{y}| \\ \leq&\epsilon \|\mathbf{w}^*\|_2, \end{aligned}$$

where m exists since $(\bar{\mathbf{x}}_t, \bar{y}_t)$ converges to $(\bar{\mathbf{x}}, \bar{y})$ according to the Law of Large Numbers.

Therefore,

$$\|\Pi_{\tau_t}(\mathbf{w}^*) - \mathbf{w}^*\|_2 \leq \epsilon. \quad \square$$

Proposition 2.2 states that, if at step m, $(\bar{\mathbf{x}}_m, \bar{y}_m)$ is close to $(\bar{\mathbf{x}}, \bar{y})$, then a sufficiently good solution (within an ϵ-ball centered at optimal solution $\mathbf{w}^*$) lies on hyper-plane τ_m. In addition, the estimation error decays and its value is upper bounded by the weighted combination of $\|\bar{\mathbf{x}}_t - \bar{\mathbf{x}}\|_2$ and $|\bar{y}_t - \bar{y}|$. Notice that CSGD optimizes the optimal empirical solution $\mathbf{w}_t^*$ that is always located on hyper-plane τ_t.

According to Theorem 2.1 and Proposition 2.2, under the assumption of regression constraint, CSGD explicitly minimizes the empirical loss as good as the second order SGD. According to Proposition 2.2, $\|\Pi_{\tau_t}(\mathbf{w}^*) - \mathbf{w}^*\|_2^2$ converges at the same rate as $\|\bar{\mathbf{x}}_t - \bar{\mathbf{x}}\|_2$ and $|\bar{y}_t - \bar{y}|$ whose exponential convergence rate [4] is supported by the Law of Large Numbers. Thus, we ignore the difference between$\Pi_{\tau_t}(\mathbf{w}^*)$ and $\mathbf{w}^*$ in our theoretical analysis for simplicity reasons.

3. A ONE-STEP DIFFERENCE INEQUALITY

To study the theoretical properties of CSGD, we start from the one-step difference bound, which is crucial to analyze the regret and convergence behavior.

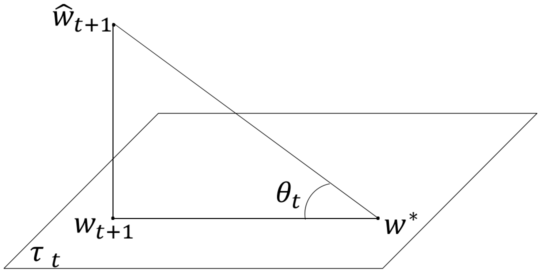

Figure 1: An illustrating example after step t. $\hat{\mathbf{w}}_{t+1}$ is the SGD result. $\mathbf{w}_{t+1}$ is the projection for $\hat{\mathbf{w}}_{t+1}$ on to the hyper-plane τ_t. $\mathbf{w}^*$ is the optimal solution. $\tan\theta_t = \|\hat{\mathbf{w}}_{t+1} - \mathbf{w}_{t+1}\|_2 / \|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2$

After iteration step t, CSGD projects the SGD result $\hat{\mathbf{w}}_{t+1}$ on the hyper-plane τ_t to get a new result $\mathbf{w}_{t+1}$ with direction and step size correction. An illustration is given in Figure 1. Note that, $\mathbf{w}^*$ is assumed on τ_t according to Proposition 2.2.

In addition, the definition of a gradient for any $\mathbf{g}_t \in \partial l(\mathbf{w}_t)$ implies

$$\begin{aligned} l(\mathbf{w}^*) &\geq l(\mathbf{w}_t) + \mathbf{g}_t^T(\mathbf{w}^* - \mathbf{w}_t) \qquad (12) \\ \Rightarrow \mathbf{g}_t^T(\mathbf{w}^* - \mathbf{w}_t) &\leq l(\mathbf{w}^*) - l(\mathbf{w}_t). \end{aligned}$$

With Eq.(12) we have the following theorems for step difference bound.

Firstly, we describe the step difference bound proved by Nemirovski for SGD.

THEOREM 3.1. (Step difference bound of SGD) *For any optimal solution $\mathbf{w}^*$, SGD has the following inequality between steps $t-1$ and t*

$$\|\hat{\mathbf{w}}_{t+1} - \mathbf{w}^*\|_2^2 - \|\hat{\mathbf{w}}_t - \mathbf{w}^*\|_2^2 \leq \eta_t^2 \|\mathbf{g}_t\|_2^2 - 2\eta_t(l(\hat{\mathbf{w}}_t) - l(\mathbf{w}^*)),$$

where η_t is the learning rate at step t.

Detail proof is given in [19].

Secondly, we prove the step difference bound of CSGD as follows.

Theorem 3.2. (Step difference bound of CSGD) *For any optimal solution* $\mathbf{w}^*$*, the following inequality holds for CSGD between steps* $t-1$ *and* t

$$\begin{aligned}&\|\mathbf{w}_{t+1}-\mathbf{w}^*\|_2^2-\|\mathbf{w}_t-\mathbf{w}^*\|_2^2 \\ \le&\eta_t^2\|\mathbf{g}_t\|_2^2-2\eta_t(l(\mathbf{w}_t)-l(\mathbf{w}^*))-\frac{\|\bar{\mathbf{g}}_{t+1}\|_2^2}{\|\bar{\mathbf{x}}_t\|_2^2},\end{aligned} \tag{13}$$

where $\mathbf{w}_t=\Pi_{\tau_t}(\hat{\mathbf{w}})$ *and* $\bar{\mathbf{g}}_{t+1}=\partial l(\hat{\mathbf{w}}_{t+1},\bar{\mathbf{x}}_t,\bar{y}_t)$*.*

Proof. Since $\hat{\mathbf{w}}_{t+1}=\mathbf{w}_t-\eta_t\mathbf{g}_t$, between two steps $t-1$ and t, $\hat{\mathbf{w}}_{t+1}$ and $\mathbf{w}_t$ follows Theorem 3.1.

Therefore, we have

$$\|\hat{\mathbf{w}}_{t+1}-\mathbf{w}^*\|_2^2-\|\mathbf{w}_t-\mathbf{w}^*\|_2^2\le\eta_t^2\|\mathbf{g}_t\|_2^2-2\eta_t(l(\mathbf{w}_t)-l(\mathbf{w}^*)). \tag{14}$$

As Euclidean projection $\mathbf{w}_{t+1}=\Pi_{\tau_t}(\hat{\mathbf{w}}_{t+1})$ given in Eq.(3), which is also shown in Figure 1, has the property,

$$\|\hat{\mathbf{w}}_{t+1}-\mathbf{w}^*\|_2^2=\|\mathbf{w}_{t+1}-\mathbf{w}^*\|_2^2+\|\mathbf{w}_{t+1}-\hat{\mathbf{w}}_{t+1}\|_2^2. \tag{15}$$

Then, substituting $\|\hat{\mathbf{w}}_{t+1}-\mathbf{w}^*\|_2^2$ given by Eq.(15) into Eq.(14) yields

$$\begin{aligned}&\|\mathbf{w}_{t+1}-\mathbf{w}^*\|_2^2-\|\mathbf{w}_t-\mathbf{w}^*\|_2^2 \\ \le&\eta_t^2\|\mathbf{g}_t\|_2^2-2\eta_t(l(\mathbf{w}_t)-l(\mathbf{w}^*))-\|\mathbf{w}_{t+1}-\hat{\mathbf{w}}_{t+1}\|_2^2.\end{aligned} \tag{16}$$

By using the projection function defined in Eq.(6), we have

$$\begin{aligned}&\|\mathbf{w}_{t+1}-\hat{\mathbf{w}}_{t+1}\|_2^2 \\ =&\|\mathbf{P}_t\hat{\mathbf{w}}_{t+1}+r_t-\hat{\mathbf{w}}_{t+1}\|_2^2 \\ =&\left\|\left(\mathbf{I}-\frac{\bar{\mathbf{x}}_t\bar{\mathbf{x}}_t^T}{\|\bar{\mathbf{x}}_t\|_2^2}\right)\hat{\mathbf{w}}_{t+1}+\frac{\bar{y}_t}{\|\bar{\mathbf{x}}_t\|_2^2}\bar{\mathbf{x}}_t-\hat{\mathbf{w}}_{t+1}\right\|_2^2 \\ =&\left\|\frac{\bar{\mathbf{x}}_t\left(\bar{y}_t-\bar{\mathbf{x}}_t^T\hat{\mathbf{w}}_{t+1}\right)}{\|\bar{\mathbf{x}}_t\|_2^2}\right\|_2^2.\end{aligned} \tag{17}$$

Since $\bar{\mathbf{g}}_{t+1}=\partial l(\hat{\mathbf{w}}_{t+1},\bar{\mathbf{x}}_t,\bar{y}_t)=-\bar{\mathbf{x}}_t\left(\bar{y}_t-\bar{\mathbf{x}}_t^T\hat{\mathbf{w}}_{t+1}\right)$, we have

$$\|\mathbf{w}_{t+1}-\hat{\mathbf{w}}_{t+1}\|_2^2=\frac{\|\bar{\mathbf{g}}_{t+1}\|_2^2}{\|\bar{\mathbf{x}}_t\|_2^2}. \quad \square$$

A direct result of the step difference bound allows the following theorem which derives the convergence result of CSGD.

Theorem 3.3. (Loss bound) *Assume (1) the norm of any gradient from* ∂l *is bounded by* G*, (2) the norm of* $\mathbf{w}^*$ *is less than or equal to* D *and (3)* $\sum_{t=1}^T\frac{\|\bar{\mathbf{g}}_{t+1}\|_2^2}{\|\bar{\mathbf{x}}_t\|_2^2}\ge G_2$ *then*

$$2\sum_{t=1}^T\eta(l(\mathbf{w}_t)-l(\mathbf{w}^*))\le D^2-G_2+G^2\sum_{t=1}^T\eta_t^2.$$

Proof. Rearranging the bound in Theorem 3.2 and sum the loss terms over t from 1 through T and then get the sum:

$$\begin{aligned}&2\sum_{t=1}^T\eta_t\left(l(\mathbf{w}_t)-l(\mathbf{w}^*)\right) \\ \le&\|\mathbf{w}_1-\mathbf{w}^*\|_2^2-\|\mathbf{w}_{T+1}-\mathbf{w}^*\|_2^2 \\ &+\sum_{t=1}^T\left(\eta_t^2\|\mathbf{g}_t\|_2^2-\frac{\|\bar{\mathbf{g}}_{t+1}\|_2^2}{\|\bar{\mathbf{x}}_t\|_2^2}\right) \\ \le&D^2-G_2+G^2\sum_{t=1}^T\eta_t^2.\end{aligned} \tag{18}$$

The final step uses the fact that $\|\mathbf{w}_1-\mathbf{w}^*\|_2^2\le D$, where $\mathbf{w}_1$ is initialized to $\mathbf{0}$, along with $\|\mathbf{g}_t\|_2^2\le G^2$ for any t and the assumption $\sum_{t=1}^T\frac{\|\bar{\mathbf{g}}_{t+1}\|_2^2}{\|\bar{\mathbf{x}}_t\|_2^2}\ge G_2$. $\square$

A corollary which is the consequence of this theorem is presented in the following. Although the convergence for CSGD follows immediately according to the Nemirovski's 3-line subgradient descent convergence proof [17], we present our first corollary underscoring the rate of convergence when η is fixed, in general is approximately $1/\epsilon^2$, or equivalently, $1/\sqrt{T}$.

Corollary 3.4. (Fixed step convergence rate) *Assume Theorem 3.3 hold and for any predetermined* T *iterations with* $\eta=\frac{1}{\sqrt{T}}$*, then*

$$\min_{t\le T}l(\mathbf{w}_t)\le\frac{1}{T}\sum_{t=1}^Tl(\mathbf{w}_t)\le\frac{1}{2\sqrt{T}}(D^2-G_2+G^2)+l(\mathbf{w}^*).$$

Proof. let $\eta_t=\eta=\frac{1}{\sqrt{T}}$ for any step t, the bound for convergence rate in Theorem 3.3 becomes,

$$2\sum_{t=1}^T(l(\mathbf{w}_t)-l(\mathbf{w}^*))\le\frac{1}{\eta}(D^2-G_2)+G^2T\eta.$$

The desired bound is achieved after plugging in the specific value of η and dividing both sides by T. $\square$

It is clear that the fixed step convergence rate for CSGD is upper bounded by SGD, which can be achieved by taking out the G_2.

4. REGRET ANALYSIS

Regret is the difference between the total loss and the optimal loss, which has been analyzed in most online algorithms for evaluating the correctness and convergence.

4.1 Regret

Let G be the upper bound of $\|\mathbf{g}_t\|_2$ for any t from $1,\ldots,T$, we have the following theorem.

Theorem 4.1. (Regret Bound for CSGD) *the regret of CSGD is:*

$$R_G(T)\le\frac{G^2}{2H}(1+\log T),$$

where H *is a constant. Therefore,*
$\limsup_{T\to\infty}R_G(T)/T\le 0$.

Proof. In Theorem 3.2, Eq.(16) shows that:

$$\begin{aligned}&2\eta_t(l(\mathbf{w}_t) - l(\mathbf{w}^*))\\ \le&\|\mathbf{w}_t - \mathbf{w}^*\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2^2\\ &+ \eta_t^2\|\mathbf{g}_t\|_2^2 - \|\mathbf{w}_{t+1} - \hat{\mathbf{w}}_{t+1}\|_2^2,\end{aligned} \tag{19}$$

where $\|\mathbf{w}_{t+1}-\hat{\mathbf{w}}_{t+1}\|_2 = \|\mathbf{w}_{t+1}-\mathbf{w}^*\|_2 \tan\theta_t$ and $\theta_t \in [0, \frac{\pi}{2})$, which is shown in Figure 1.

Therefore, sum Eq.(19) over t from 1 to T, we have

$$\begin{aligned}&2\sum_{t=1}^{T} l(\mathbf{w}_t) - l(\mathbf{w}^*)\\ \le&\sum_{t=2}^{T} \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \frac{\tan^2\theta_{t-1}}{\eta_{t-1}}\right)\\ &+ \frac{1}{\eta_1}\|\mathbf{w}_1 - \mathbf{w}^*\|_2^2 + G^2\sum_{t=1}^{T}\eta_t.\end{aligned} \tag{20}$$

By adding a dummy term $-\frac{1}{\eta_0}(1+\tan^2\theta_0)\|\mathbf{w}_1-\mathbf{w}^*\|_2^2 = 0$ on the right side, we have

$$\begin{aligned}&2\sum_{t=1}^{T} l(\mathbf{w}_t) - l(\mathbf{w}^*)\\ \le&\sum_{t=1}^{T} \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \frac{\tan^2\theta_{t-1}}{\eta_{t-1}}\right) + G^2\sum_{t=1}^{T}\eta_t.\end{aligned} \tag{21}$$

Note that, $\tan\theta_t$ does not decrease w.r.t step t as shown in Lemma 4.2 and η_t does not increase w.r.t step t.

Since $\eta_{t-1} > 0$, we assume the lower bound $\frac{\tan^2\theta_{t-1}}{\eta_{t-1}} \ge H$ hold [1], then Eq.(21) can be rewritten as

$$\begin{aligned}&2\sum_{t=1}^{T} l(\mathbf{w}_t) - l(\mathbf{w}^*)\\ \le&\sum_{t=1}^{T} \|\mathbf{w}_t - \mathbf{w}^*\|_2^2 \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - H\right) + G^2\sum_{t=1}^{T}\eta_t.\end{aligned} \tag{22}$$

Set $\eta_t = \frac{1}{Ht}$ for all $t \ge 1$, we have

$$\sum_{t=1}^{T} l(\mathbf{w}_t) - l(\mathbf{w}^*) \le \frac{G^2}{2H}(1 + \log T). \quad \square \tag{23}$$

When $\tan\theta_t$ becomes small, the improvement from the constraint will not be significant as shown in Figure 1. Lemma 4.2 shows that $\tan\theta_t$ does not decrease as t increases if $\hat{\mathbf{w}}_{t+1}$ and $\mathbf{w}_{t+1}$ are close to $\mathbf{w}^*$. This indicates the improvement from $\hat{\mathbf{w}}_{t+1}$ to $\mathbf{w}_{t+1}$ is stable under the regression constraint. And this further proves the stableness of the regret bound.

Lemma 4.2. $\tan\theta_t = \|\mathbf{w}_{t+1}-\hat{\mathbf{w}}_{t+1}\|_2/\|\mathbf{w}_{t+1}-\mathbf{w}^*\|_2$ *does not decrease w.r.t step t.*

Proof. It is known that $\hat{\mathbf{w}}_{t+1}$ and $\mathbf{w}_{t+1}$ converge to $\mathbf{w}^*$. If $\mathbf{w}_{t+1}$ converges beyond the speed of $\hat{\mathbf{w}}_{t+1}$, $\tan\theta_t$ will diverge and the Lemma holds for sure.

SGD has the convergence rate $O(1/\sqrt{T})$. This is the worst convergence rate that can be obtained by all the stochastic optimization approaches. Before we prove CSGD has a faster convergence rate than SGD, we temporarily make a conservative assumption that CSGD and SGD both converge at the rate of $O(t^{-\alpha})$, where α is a positive number.

Let $\|\hat{\mathbf{w}}_{t+1}-\mathbf{w}^*\|$ be $at^{-\alpha}$ and $\|\mathbf{w}_{t+1}-\mathbf{w}^*\|$ be $bt^{-\alpha}$. Along with Eq.(15), we have

$$\begin{aligned}\tan\theta_t &= \|\mathbf{w}_{t+1} - \hat{\mathbf{w}}_{t+1}\|_2/\|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2\\ &= \frac{\sqrt{\|\hat{\mathbf{w}}_{t+1} - \mathbf{w}^*\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2^2}}{\|\mathbf{w}_{t+1} - \mathbf{w}^*\|_2}\\ &= \frac{\sqrt{a^2 - b^2}}{b}\\ &= O(1). \quad \square\end{aligned}$$

In our approach, the $O(\log T)$ regret bound achieved by CGSD neither requires strong convexity nor regularization, while Hazan et al. achieve the same $O(\log T)$ regret bound under the assumption of strong convexity [10], and Bartlett et al. use regularization to obtain the same $O(\log T)$ regret bound for strongly convex functions and $O(\sqrt{T})$ for any arbitrary convex functions. Furthermore, the $O(\log T)$ regret bound of CGSD is better than the general regret bound $O(\sqrt{T})$ discussed in [30].

The regret bound suggests a decreasing step size, which yields the convergence rate stated in the following corollary.

Corollary 4.3. (Decreasing step convergence rate) *Assume Theorem 4.1 hold and $\eta_t = \frac{1}{Ht}$ for any step t, then*

$$\min_{t\le T} l(\mathbf{w}_t) \le \frac{1}{T}\sum_{t=1}^{T} l(\mathbf{w}_t) \le \frac{G^2}{2TH}(1 + \log T) + l(\mathbf{w}^*).$$

This corollary is a direct result of Theorem 4.1. It shows that the $O(\log T/T)$ convergence rate of CSGD is much better than $O(1/\sqrt{T})$ obtained by SGD.

4.2 Learning rate strategy

Theorem 4.1 and Corollary 4.3 suggest an optimal learning rate decreasing at the rate of $O(1/t)$ without assuming the strong convexity for the objective function. However, the decay proportional to the inverse of the number of iterations is not robust to the wrong setting of the proportionality constant. The typical result for the wrong proportionality constant will lead to divergence in the first several iterations or converge to a point far away from the optimal. Motivated by this problem, we propose a 2-phase learning rate strategy, which is defined as

$$\eta_t = \begin{cases}\eta_0/\sqrt{t}, & t < m\\ \eta_0\sqrt{m}/t, & t \ge m\end{cases}.$$

The step m is achieved when desired error tolerance ϵ is obtained in Proposition 2.2, $m = O(1/\epsilon)$. The maximum value for m is the total size of the dataset, since the global mean would be achieved after one pass of the data.

[1] This inequality is based on the assumption that H is positive. Although this assumption could be slightly violated when $\tan\theta_t = 0$ if $\mathbf{w}_t$ lies on τ_t and $(\mathbf{x}_t, y_t) = (\bar{\mathbf{x}}_t, \bar{y}_t)$, this event rarely happens in real cases. Even if it happens but for finite times, the legality of our analysis is still provable. So we simply rule out this rare event in theoretical analysis.

5. EXPERIMENTS

5.1 Optimization study

In this section, we perform numerical simulation to systematically analyze the proposed algorithms and conduct empirical verification of our theoretical results.

Our optimization study includes 5 algorithms, including the proposed CSGD and NCSGD, and SGD, ASGD, 2SGD, for comparative study:

1) Stochastic Gradient Descent (SGD) (a.k.a Robbins-monro algorithm) [12]: SGD, which is also known as the Least Mean Squares approach to solve the stochastic least squares, is chosen as the baseline algorithm.

2) Averaged Gradient Descent (ASGD) (a.k.a. Polyak-Ruppert averaging) [21]: ASGD performs the SGD approach and returns the average point at each iteration. ASGD, as the state-of-the-art approach on first order stochastic optimization, has achieved good performance [29, 2].

3) Constrained Stochastic Gradient Descent (CSGD): CSGD uses the 2-phase learning rate strategy in order to achieve the proposed regret bound.

4) Naive Constrained Stochastic Gradient Descent (NCSGD): NCSGD is a naive version of CSGD, which updates with the same learning rate of SGD to illustrate the optimization error of NCSGD is upper bounded by SGD at each step.

5) Second Order Stochastic Gradient Descent (2SGD) [7]: 2SGD replaces the learning rate η by the inverse of the Hessian matrix, which also forms Recursive Least Squares, a second order stochastic least squares solution. Compared to the first order approaches, 2SGD is considered as the best possible approach using a full Hessian matrix.

The experiments for least squares optimization have been conducted on two different settings: strongly and non-strongly convex cases. The difference between strongly convex and non-strongly convex objectives has been extensively studied in convex optimization and machine learning on selection of the learning rate strategy [2, 24]. Even though a decay of the learning rate at the rate of the inverse of the number of samples has been theoretically suggested to achieve the optimal rate of convergence in strongly convex case [10]. In practice, the least squares approach may decrease too fast and the iteration will "get stuck" too far away from the optimal solution [12]. To solve this problem, a slower decay has been proposed in [2] for learning rate, $\eta_t = \eta_0 t^{-\alpha}$ and $\alpha \in (1/2, 1)$. A $\lambda/2\|\mathbf{w}\|_2^2$ regularization term can also be added to obtain λ-strongly convex and uniformly Lipschitz [29, 1]. To ensure convergence, we safely set $\alpha = 1/2$ [30] as a robust learning rate for all algorithms in our empirical study, which guarantees all algorithms can converge in close vicinity to the optimal solution. In order to study the real convergence performance of different approaches, we use the prior knowledge of the spectrum of the Hessian matrix to obtain the best η_0. To achieve the regret bound in Theorem 4.1, a 2-phase learning rate is utilized for CSGD and m is set to be the half of the dataset size $n/2$.

We generate n input samples with d dimensions $i.i.d.$ from a uniform distribution between 0 and 1. The optimal coefficient $\mathbf{w}^*$ is randomly drawn from standard Gaussian distribution. Then the n data samples for our experiments is constructed using the n input samples as well as the coefficient with a zero-mean Gaussian noise with variance 0.2. Two settings for least squares optimization are designed: 1) a low-dimension case with strong convexity, where n =10,000, d =100, 2) a high-dimension case, where n =5,000, d =5,000, with smallest eigenvalue of the Hessian matrix close to 0, which yields a non-strongly convex case. In each iteration round, one sample is randomly drawn from one individual dataset using a uniformly distribution.

In the strongly convex case, as shown in Figure 2 top row, CSGD behaves similar to 2SGD and outperforms other first order approaches. As we know, 2SGD, as a second order approach, is the best possible solution per iteration for all first order approaches. However, 2SGD requires $O(d^2)$ computation and space complexity in each iteration. CSGD performs like 2SGD but only requires $O(d)$ computation and space complexity, and CSGD has a comparable performance as 2SGD when doing optimization by giving a certain amount of CPU time, as shown in top right slot of Figure 2.

In the non-strongly convex case, as shown in Figure 2 bottom row, CSGD performs the best among all first order approaches. 2SGD becomes impractical in this high dimensional case due to its high computation complexity. CSGD has a slow start at the beginning and this is because it adopts a larger initial learning rate η_0 which yields better convergence for the second phase. This fact is also consistent with the comparisons using the wrong initial learning rates discussed in [2]. However, this larger initial learning rate speeds up the convergence in the second phase. In addition, the non-strong convexity corresponds to an infinite ratio of the eigenvalues for the Hessian matrix, which significantly slows the performance of SGD. NCSGD has not been influenced by this case and still performs consistently better than SGD.

In our empirical study, we have observed: 1) NCSGD consistently performs better than SGD, and the experimental results verify Corollary4.3; 2) CSGD performs very similarly to the second order approaches, and this supports the regret bound discussed in Theorem 4.1; 3)CSGD performs the best among all the other state-of-the-art first order approaches ; 4) CSGD is the most competent algorithm to deal with high-dimension data among all the other algorithms in our experiments.

5.2 Classification extension

In the classification task, least squares loss function plays an important role and the optimization of least squares is the cornerstone of all least squares based algorithms, such as Least Squares Support Vector Machine (LS-SVM) [26], Regularized Least-Squares classification [22, 6] and etc. In this case, SGD is utilized by default during the optimization.

It is well acknowledged that the optimization speed for least squares directly affects the performance of least squares based algorithms. A faster optimization procedure corresponds to less training iterations and less training time. Therefore, replacing the SGD with CSGD for the least squares optimization in many algorithm, the performance could be greatly improved. In this subsection, we show an example

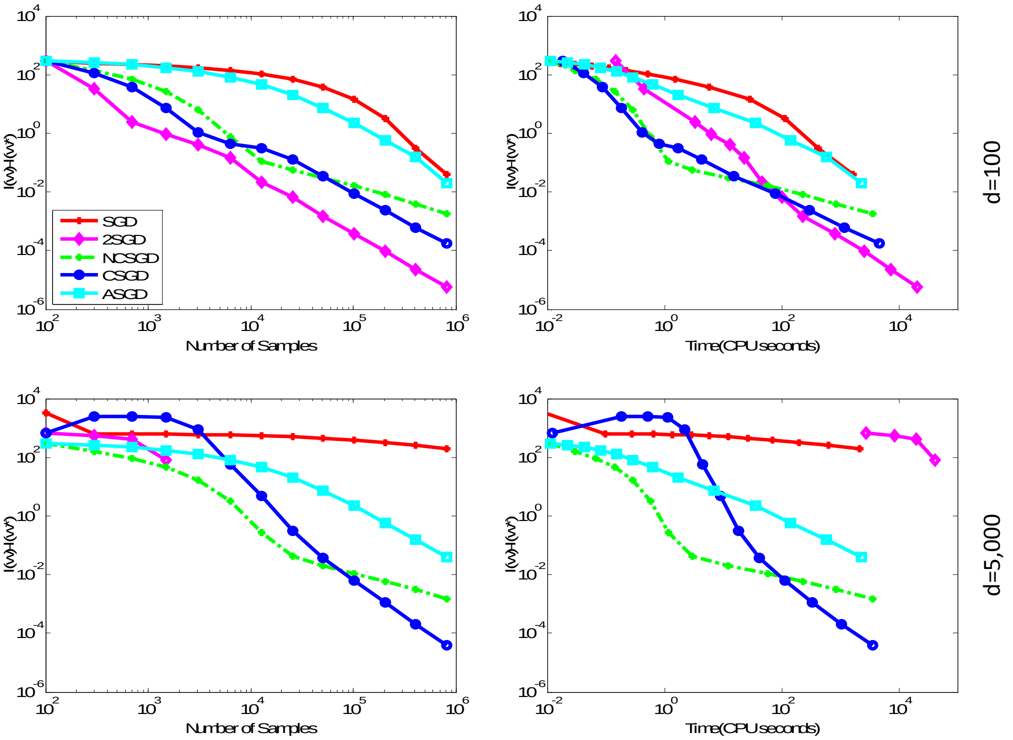

Figure 2: Comparison of the methods on the low-dimension case (top), and the high-dimension case (bottom).

how to adopt CSGD in the optimization for the existing classification approaches.

One direct classification approach using least squares loss is ADAptive LINear Element (Adaline) [16], which is a well-known method in neural network. Adaline adopts a simple perceptron-like system that accomplishes classification, which modifies coefficients to minimize the least squares error at every iteration. Note that, although it may not achieve a perfect classification by using a linear classifier, the direct optimization for least squares is commonly used as a subroutine in many complex algorithms, such as Multiple Adaline (Madaline) [16] to achieve the non-linear separability by using multiple layers of Adalines. Since the fundamental optimization procedures for these least squares algorithms are the same, we only show a basic case for Adaline to show CSGD can improve the performance.

In Adaline, each neuron separates two classes using a coefficient vector $\mathbf{w}$. The equation of the separating hyper-plane can be derived from the coefficient vector. Specifically, to classify input sample $\mathbf{x}_i$, let net be the net input of this neuron, where $net = \mathbf{w}^T\mathbf{x}_t$. The output of Adaline o_t is 1 when $net > 0$ and o_t is -1 otherwise.

The crucial part for training the Adaline is to obtain the best coefficient vector $\mathbf{w}$, which is updated per iteration by minimizing the squared error. At iteration t, the squared error is $\frac{1}{2}(y_t - net_t)^2$, where y_t is 1 or -1 representing the positive class or negative class respectively. Adaline adopts SGD for optimization whose learning rule is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\mathbf{g}_t, \tag{24}$$

where η is a constant learning rate, and the gradient $\mathbf{g}_t = -(y_t - \mathbf{w}_t^T\mathbf{x}_t)\mathbf{x}_t$.

When replacing SGD with CSGD for Adaline, Eq.(24) is replaced with Eq.(9).

In the multiclass classification case, suppose there are c classes, Adaline needs c neurons to perform the classification and each neuron still performs the binary class discrimination. The CSGD version of Adaline (C-Adaline) is depicted in Algorithm 2, which is straightforward and easy to implement. One thing need to be pointed out is that, the class label y_t has to be rebuilt in order to fit c neurons. Therefore, the class label y_t of sample $\mathbf{x}_t$ for neuron c_i is defined as: $y_{(t,c_i)} = 1$ when $y_t = i$ and $y_{(t,c_i)} = 0$ otherwise. The output $o_t = k$ means that the k^{th} neuron c_k has the highest net value among c neurons.

Algorithm 2 CSGD version of Adaline (C-Adaline)

```
Initialize $\mathbf{w}_0 = \bar{\mathbf{x}}_0 = \mathbf{0}$ and $\bar{y}_0 = 0$.
for $t = 1, 2, 3, \ldots$ do
  for $i = 1, 2, 3, \ldots, c$ do
    Compute the gradient $\mathbf{g}_t \in \partial l(\mathbf{w}_t, \mathbf{x}_t, y_{(t,c_i)})$.
    Compute $(\bar{\mathbf{x}}_t, \bar{y}_{(t,c_i)})$ with
    $\bar{\mathbf{x}}_t = \frac{t-1}{t}\bar{\mathbf{x}}_{t-1} + \frac{1}{t}\mathbf{x}_t$, and
    $\bar{y}_{(t,c_i)} = \frac{t-1}{t}\bar{y}_{(t-1,c_i)} + \frac{1}{t}y_{(t,c_i)}$.
    $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\mathbf{g}_t - \bar{\mathbf{x}}_t\left(\bar{\mathbf{x}}_t^T(\mathbf{w}_t - \eta_t\mathbf{g}_t)\right)/\|\bar{\mathbf{x}}_t\|_2^2 + \mathbf{r}_t$.
  end for
end for
```

To evaluate the improvement of the C-Adaline,we provide computational experiments of both Adaline and C-Adaline on the MNIST dataset [14], which is widely used as stochastic optimization classification benchmark on handwritten digits recognition [9, 28].

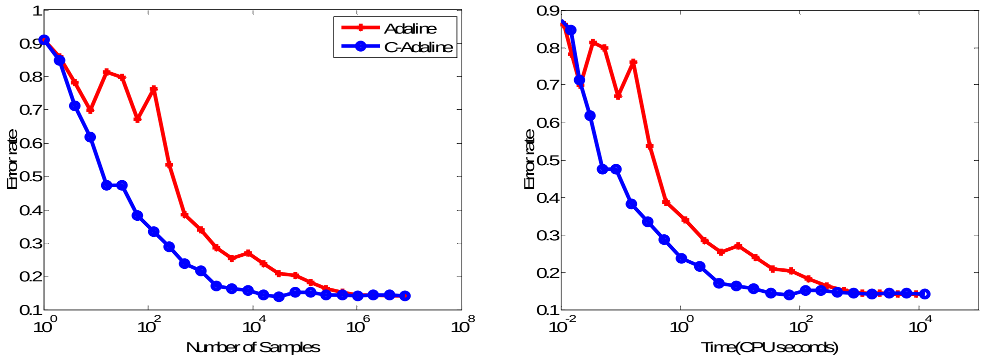

Figure 3: Left: test error rate versus iteration on the MNIST classification. Right: test error rate versus CPU time on the MNIST classification. C-Adaline is the proposed CSGD version of Adaline

The MNIST dataset consists of 60,000 training samples and 10,000 test samples with 10 classes. Each digit is presented by a 28 × 28 gray scale image, for a total of 784 features. All the data is downscaled to $[0, 1]$ via dividing the maximum pixel intensity by 255. For the setting of the learning rate, Adaline adopts a constant while C-Adaline simply takes the updating rule of Naive Constrained Stochastic Gradient Descent (NCSGD). η for Adaline is set to 2^{-17}, and the C-Adaline has $\eta_0 = 2^{-4}$, which are both the optimal results selected from $2^{-20}, 2^{-19}, \cdots, 2^{-1}$. Since the optimal solution is unique, this experiment is to examine how fast can Adaline and C-Adaline converge to this optimal.

The test set error rate as a functions of number of operations is shown in Figure 3 (Left). It is clear that both Adaline and C-Adaline converge to the same test error because, they both optimize the least squares error. C-Adaline achieves 0.14 test error after processing 2^{14} samples ($\approx 10^4$), while Adaline achieves 0.14 test error after processing 2^{20} samples ($\approx 10^6$). This indicates that C-Adaline converges 64 times as fast as Adaline! Considering the size of the training set is 60,000, C-Adaline uses 1/4 of the total training samples to achieve the nearly optimal test error rate, while Adaline needs to visit each training sample more than 16 times to achieve the same test error rate. Figure 3 (Right) shows the test error rate versus the CPU time. To achieve the 0.14 test error, Adaline consumes 112.47 seconds, while C-Adaline only takes 3.38 seconds. Note that, in this experiment, 10 neurons are trained in parallel. It is another achievement to get the nearly optimal test error using a least squares classifier in about 3 seconds for a 10^6 scale dataset.

To better understand the classification results, in Figure 4, we visualize the data samples on a two dimensional space by t-SNE [27], which is a nonlinear mapping commonly used for exploring the inherent structure from high dimensional data. Since a linear classifier does not perform well on this problem and both algorithms have the same classification error ultimately, we suppress the samples which are still misclassified in Figure 4 for clarity's sake. When both algorithms have processed 2^{14} training samples, their classification results on 10,000 test samples are depicted in Figure 4. C-Adaline misclassified 212 samples while Adaline misclassified 1248 samples, which is about 6 times as C-Adaline.

This experiment compares a SGD based classifier (Adaline) and the proposed CSGD improvement version (C-Adaline) using the MNIST dataset. In summary, C-Adaline consistently has a better test error rate than the original Adaline during the optimization. For a given test error rate, C-Adaline takes much less CPU time than Adaline.

6. CONCLUSION

In this paper, we analyze a new constrained based stochastic gradient descent solution for the large-scale least square problem. We provide theoretical justifications for the proposed method, called CSGD and NCSGD, which utilize the averaging hyper-plane as the projected hyper-plane. Specifically, we described the convergence rates as well as the regret bounds for the proposed method. CSGD performs like a full second order approach but with simpler computation than 2SGD. The optimal regret $O(\log T)$ is achieved in CSGD when adopting a corresponding learning rate strategy. The theoretical superiorities are justified by experimental results. In addition, it is easy to extend the SGD based least squares algorithms to CSGD and the CSGD version can yield better performance. An example of upgrading Adaline from SGD to CSGD is used to demonstrate the straightforward but efficient implementation of CSGD

References

[1] A. Agarwal, S. Negahban, and M. Wainwright. Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions. *Advances in Neural Information Processing Systems*, 2012.

[2] F. Bach and E. Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

[3] P. L. Bartlett, E. Hazan, and A. Rakhlin. Adaptive online gradient descent. *Advances in Neural Information Processing Systems*, 2007.

[4] L. E. Baum and M. Katz. Exponential convergence rates for the law of large numbers. *Transaction American Mathematical Society*, pages 771–772, 1963.

[5] D. P. Bertsekas. Nonlinear programming. *Athena Scientific*, 1999.

[6] C. M. Bishop. Pattern recognition and machine learning. *Springer-Verlag New York*, 2006.

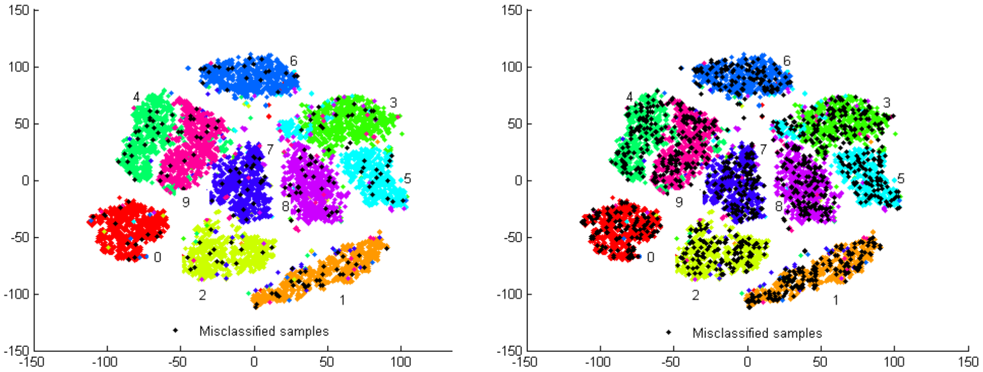

Figure 4: Two dimensional t-SNE visualization of the classification results for C-Adaline (Left) and Adaline (Right) on MNIST dataset when 2^{14} samples have been processed.

[7] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. *Advances in Neural Information Processing Systems*, 20:161–168, 2008.

[8] L. Bottou and Y. LeCun. Large scale online learning. *Advances in Neural Information Processing Systems*, 2003.

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

[10] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Conference on Learning Theory*, 69(2-3):169–192, Dec. 2007.

[11] C. Hu, J. T. Kwok, and W. Pan. Accelerated gradient methods for stochastic optimization and online learning. *Advances in Neural Information Processing Systems*, pages 781–789, 2009.

[12] H. J. Kushner and G. Yin. Stochastic approximation and recursive algorithms and applications. *Springer-Verlag*, 2003.

[13] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, June 2009.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] L. Ljung. Analysis of stochastic gradient algorithms for linear regression problems. *IEEE Transactions on Information Theory*, pages 30(2):151–160, 1984.

[16] K. Mehrotra, C. K. Mohan, and S. Ranka. *Elements of artificial neural networks*. MIT press, 1996.

[17] A. Nemirovski. Efficient methods in convex programming. *Lecture Notes*, 1994.

[18] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, Jan. 2009.

[19] A. Nemirovski and D. Yudin. Problem complexity and method efficiency in optimization. *John Wiley and Sons Ltd*, 1983.

[20] Y. Nesterov. Introductory lectures on convex optimization. *A basic course(Applied Optimization)*, 2004.

[21] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, pages 838–855, 1992.

[22] R. Rifkin, G. Yeo, and T. Poggio. Regularized least-squares classification. *Nato Science Series Sub Series III Computer and Systems Sciences*, 190:131–154, 2003.

[23] S. Shalev-Shwartz and S. M. Kakade. Mind the duality gap: Logarithmic regret algorithms for online optimization. *Advances in Neural Information Processing Systems*, pages 1457–1464, 2008.

[24] S. Shalev-Shwartz and N. Srebro. Svm optimization: inverse dependence on training set size. *International Conference on Machine Learning*, 2008.

[25] J. C. Spall. Introduction to stochastic search and optimization. *John Wiley and Sons, Inc*, 2003.

[26] J. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 1999.

[27] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[28] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *The Journal of Machine Learning Research*, 11:2543–2596, 2010.

[29] T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. *International Conference on Machine Learning*, 2004.

[30] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. *International Conference on Machine Learning*, 2003.