Consistently better than SGD

Best regret bound

Ready to use

Code: http://www.cs.umb.edu/~yangmu/code/CSGD.zip

Constrained Stochastic Gradient Descent for Large-scale Least Squares Problem

Yang Mu[1], Wei Ding[1], Tianyi Zhou[2], Dacheng Tao[2]

[1]University of Massachusetts Boston

[2]University of Technology, Sydney

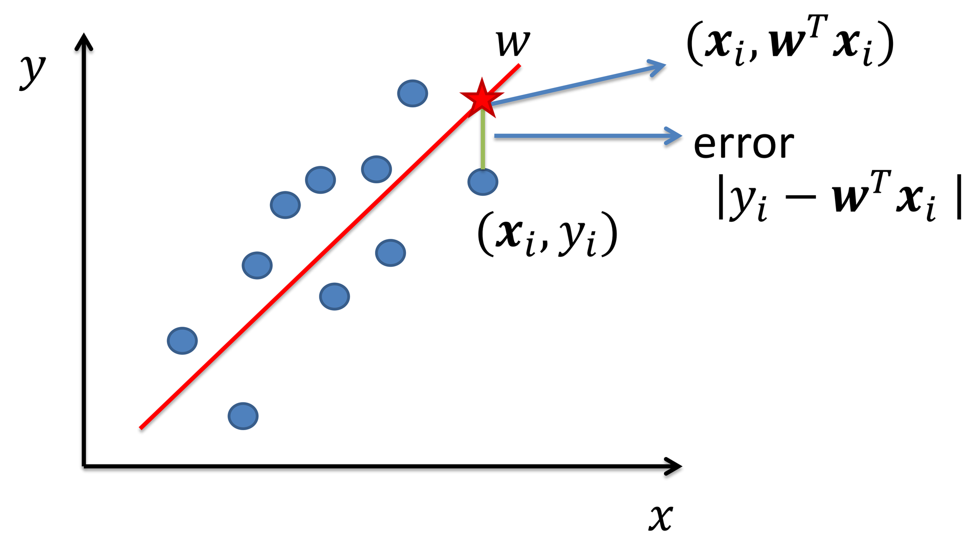
University of Massachusetts UMASS Boston

UTS University of Technology, Sydney

1. The Least Squares problem

Objective function:

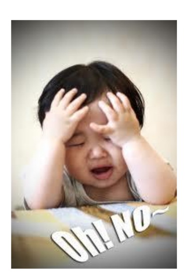
$\min_{\boldsymbol{w}} \sum_i \frac{1}{2}\|y_i - \boldsymbol{w}^T\boldsymbol{x}_i\|_2^2$, where $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}^1$ and $\boldsymbol{w} \in \mathbb{R}^d$



Linear regression

2. When Least Squares problem meets Large Scale

Use closed form solution: $\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$?



It's $\boldsymbol{O}(\boldsymbol{nd}^2)$!

Note: Any algorithm with time complexity greater than $\boldsymbol{O}(\boldsymbol{nd})$ is not applicable in large scale high dimension cases.

Stochastic Gradient Descent (**SGD**) with $\boldsymbol{O}(\boldsymbol{d})$ time each iteration is an appealing approach, which takes the form

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

where $l(\boldsymbol{w}, \boldsymbol{x}_i, y_i) = \frac{1}{2}\|y_i - \boldsymbol{w}^T\boldsymbol{x}_i\|_2^2$ is the loss function at step i, denoted as $l(\boldsymbol{w})$ for short.

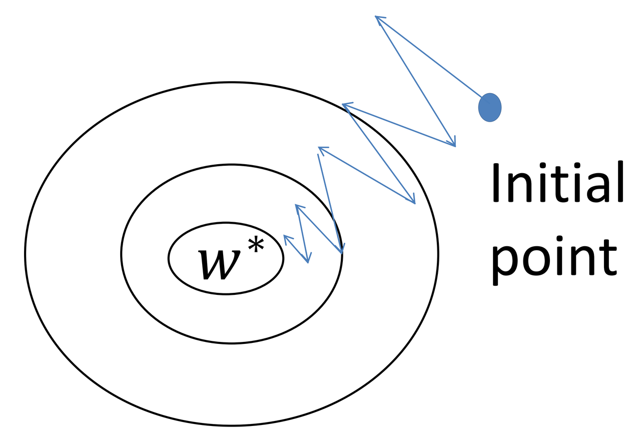
SGD update rule is:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t\boldsymbol{g}_t,$$

where $\boldsymbol{g}_t = \partial l(\boldsymbol{w})$.

3. Motivation

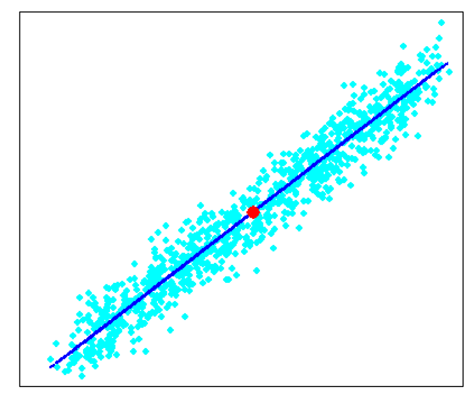
Zig-Zag property of SGD



Slow!

\+

The beauty of Least Squares

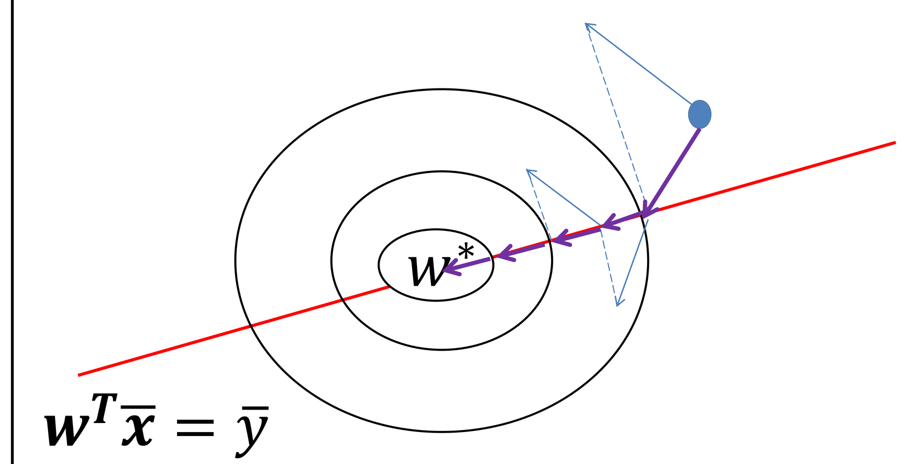


Optimal passes the mean, $\boldsymbol{w}^{*T}\bar{\boldsymbol{x}} = \bar{y}$

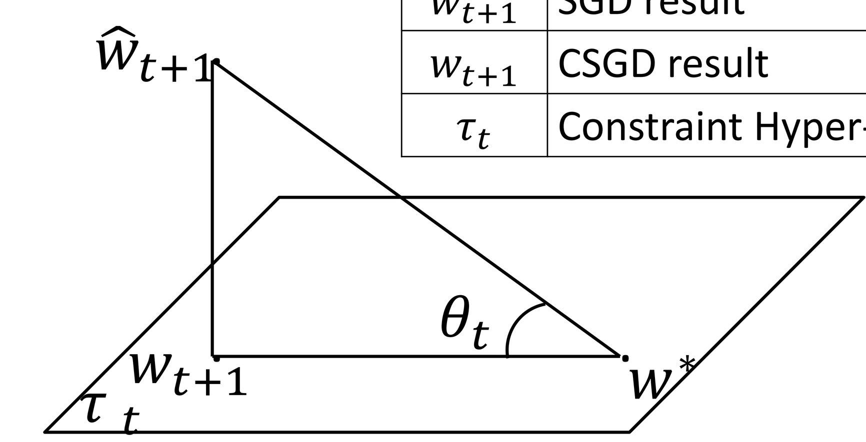
⇒

Constrained Stochastic Gradient Descent (CSGD)

w^*	batch optimal
$\hat{w}_{t+1}$	SGD result
w_{t+1}	CSGD result
τ_t	Constraint Hyper-plane



Use mean constraint to leverage SGD



Projection close view in each step

Remarks:

1. performs **consistently better** than SGD in terms of batch optimal. 2. has $\boldsymbol{O}(\log \boldsymbol{T})$ regret bound $R_G(T) \le \frac{G^2}{2H}(1 + \log T)$ 3. **extensible.**

4. Algorithm

CSGD:

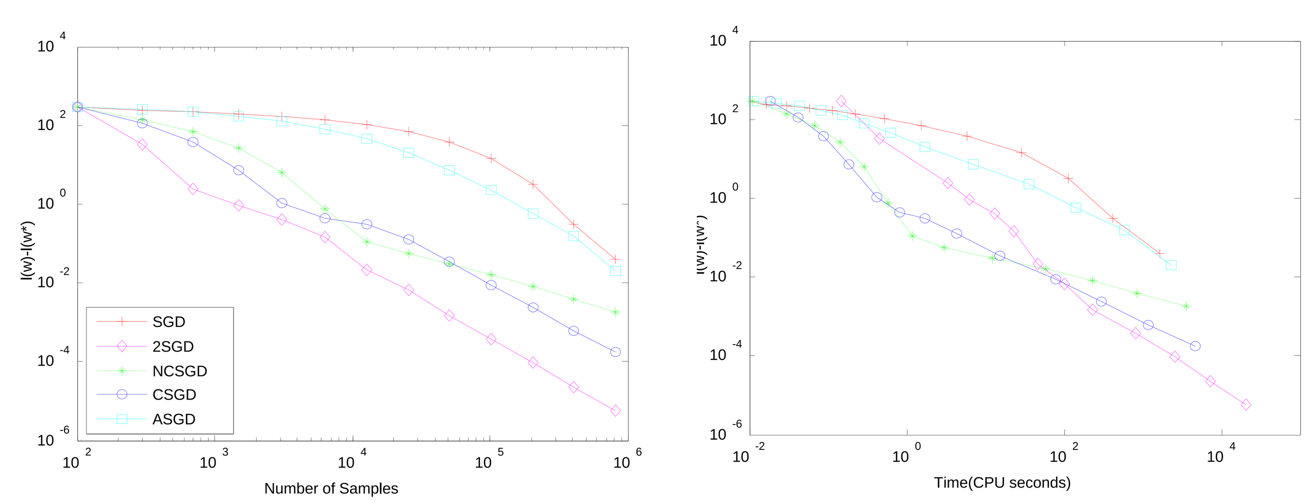
$\boldsymbol{w}_{t+1}^* = \arg\min_{\boldsymbol{w}} \frac{1}{t}\sum_{i=1}^{t}\frac{1}{2}\|y_i - \boldsymbol{w}^T\boldsymbol{x}_i\|_2^2$, s.t. $\boldsymbol{w}^T\bar{\boldsymbol{x}}_t = \bar{y}_t$. The update rule $\boldsymbol{w}_{t+1} = \mathbf{P}_t(\boldsymbol{w}_t - \eta_t\boldsymbol{g}_t) + \boldsymbol{r}_t$, where $\mathbf{P}_t = I - \frac{\bar{\boldsymbol{x}}_t\bar{\boldsymbol{x}}_t^T}{\|\bar{\boldsymbol{x}}_t\|_2^2}$, $r_t = \frac{\bar{y}_t}{\|\bar{\boldsymbol{x}}_t\|_2^2}\bar{\boldsymbol{x}}_t$

WAIT! Is $\mathbf{P_t} \in \mathbb{R}^{d\times d}$? **YES**, but it is rank ONE!

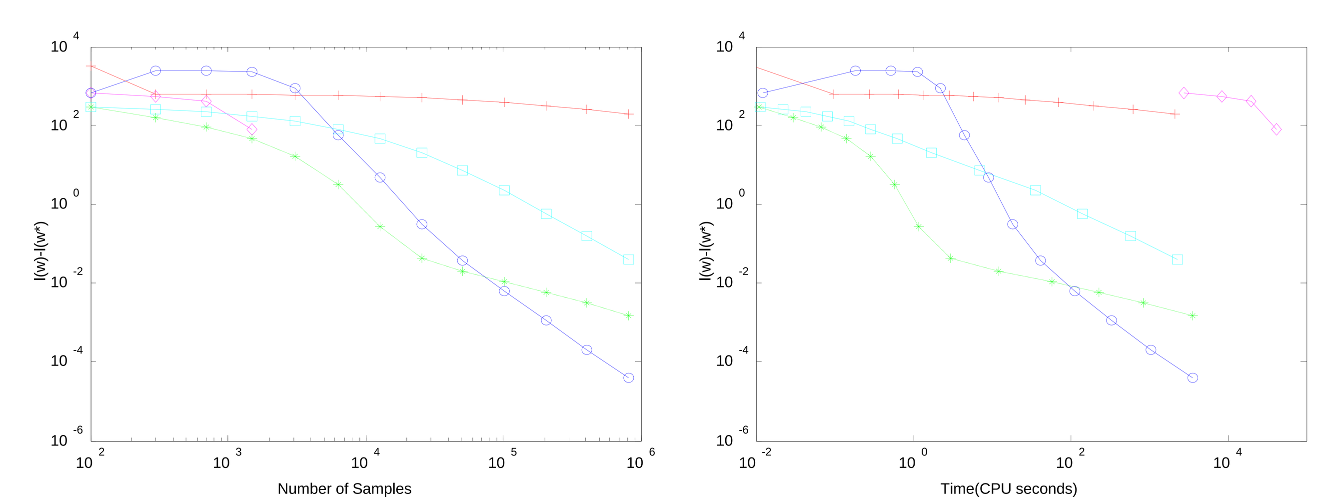
Therefore, we still have a $\boldsymbol{O}(\boldsymbol{d})$ time complexity algorithm each iteration: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t\boldsymbol{g}_t - \bar{\boldsymbol{x}}_t\left(\bar{\boldsymbol{x}}_t^T(\boldsymbol{w}_t - \eta_t\boldsymbol{g}_t)\right)/\|\bar{\boldsymbol{x}}_t\|_2^2 + \boldsymbol{r}_t$.

5. Experiments

Optimization Study



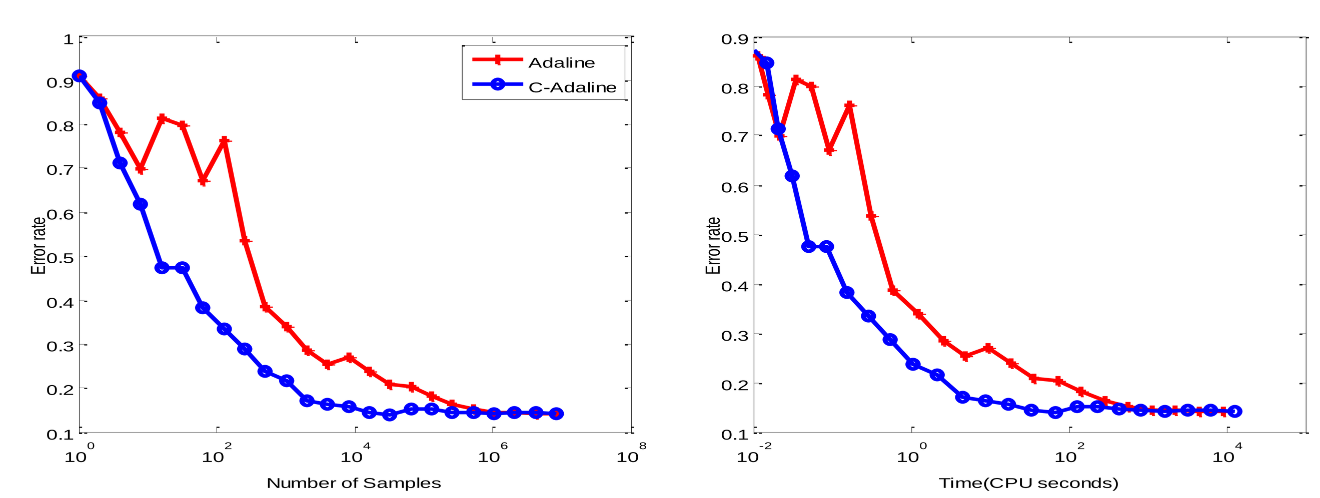
d=100, strong convex

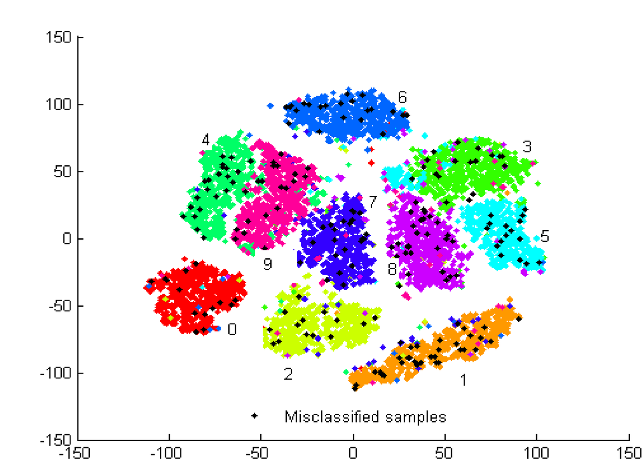


d=5,000, non strong convex

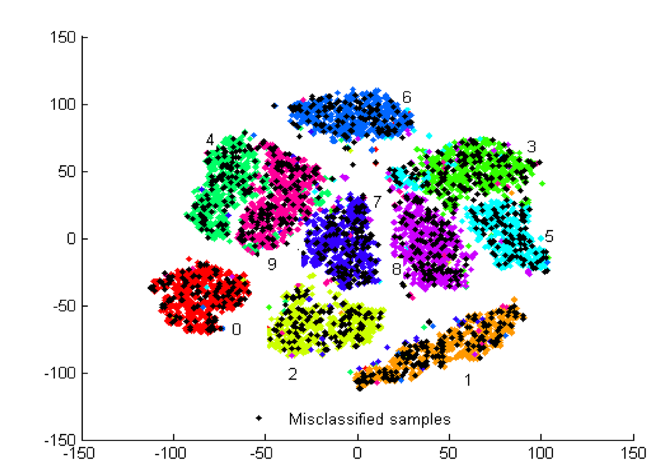
Synthetic dataset

Classification Study





212 misclassified samples in 3.38 seconds



1248 misclassified samples in 112.47 seconds

MNIST dataset