Convex Optimization

Part 6: Variance reduction methods

Namhoon Lee[1]

POSTECH

7 Dec 2022

[1] Slides courtesy of Donghyun Oh, Jinseok Chung

Table of Contents

Table of Contents

Problem Setup

Many machine learning problem involves optimizing the following problem

$$\min_{x \in \mathbb{R}^p} g(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

where each f_i is smooth and convex. Often we deal with cases where g is strongly convex.

For optimization, gradient descent (GD, full gradient) method iterates by

$$x^{k+1} = x^k - \alpha_k \nabla g(x^k) = x^k - \frac{\alpha_k}{n} \sum_{i=1}^{n} \nabla f_i(x^k)$$

Stochastic gradient descent (SGD) method iterates by

$$x^{k+1} = x^k - \alpha_k \nabla f_{i_k}(x^k)$$

where i_k is sampled uniformly from $\{1, ..., n\}$.

GD vs SGD

For GD, suboptimality bound at iteration k, using constant α_k, is given as

$$g(x^k) - g(x^*) = O(1/k)$$

if each f_i is smooth and convex.

$$g(x^k) - g(x^*) = O(\rho^k), \rho < 1$$

if in addition, g is strongly convex.

For SGD, suboptimality bound at iteration k, using decreasing α_k, is given as

$$\mathbb{E}[g(x^k)] - g(x^*) = O(1/\sqrt{k})$$

if each f_i is smooth and convex.

$$\mathbb{E}[g(x^k)] - g(x^*) = O(1/k)$$

if in addition, g is strongly convex.

SAG

Want to have an algorithm with the low cost of SGD, and convergence rate of GD, with constant step size!

Stochastic average gradient (SAG) algorithm proceeds as follows:

$$x^{k+1} = x^k - \frac{\alpha_k}{n} \sum_{i=1}^{n} y_i^k$$

where

$$y_i^k = \begin{cases} \nabla f_i(x^k), & i = i_k \\ y_i^{k-1}, & \text{else} \end{cases}$$

Here y_i^k is an estimate of $\nabla f_i(x^k)$ for each data i. Has access to i_k and keeps a memory of the recent gradient value computed for each i.

Convergence analysis

Assumptions

1. Each f_i is convex and differentiable
2. Each gradient of f_i, ∇f_i is Lipschitz constant with constant L, that is $\|\nabla f_i(x) - \nabla f_i(y)\|_2 \leq L\|x - y\|_2$
3. There is a minimizer x^* of g.
4. (For 2nd part of theorem) g is strongly convex with constant $\mu > 0$, that is $g(x) - \frac{\mu}{2}\|x\|_2^2$ is convex.

Notations

1. average iterate $\bar{x}^k = \frac{1}{k}\sum_{i=0}^{k-1} x^i$
2. variance of gradient norms at the optimum $\sigma^2 = \frac{1}{n}\sum_{i=0}^{k-1}\|\nabla f_i(x^*)\|_2^2$

Theorem (convex case)

With a constant step size $\alpha_k = \frac{1}{16L}$, the SAG iterations for $k \geq 1$ satisfy

$$\mathbb{E}[g(\bar{x}^k)] - g(x^*) \leq \frac{32n}{k} C_0$$

Theorem (strongly convex case)

Further, if g is μ-strongly convex, we have

$$\mathbb{E}[g(x^k)] - g(x^*) \leq \left(1 - \min\{\frac{\mu}{16L}, \frac{1}{8n}\}\right)^k C_0$$

- Here, if we initialize with $y_i^0 = 0$, we have

$$C_0 = g(x^0) - g(x^*) + \frac{4L}{n}\|x^0 - x^*\|_2^2 + \frac{\sigma^2}{16L}$$

- and if we initialize with $y_i^0 = \nabla f_i(x^0) - \nabla g(x^*)$, we have

$$C_0 = \frac{3}{2}[g(x^0) - g(x^*)] + \frac{4L}{n}\|x^0 - x^*\|_2^2$$

Proof Outline

- For strongly convex case $\mu \geq 0$, consider a Lyapunov function of the form

$$\mathcal{L}(\theta^k) = 2hg(x^k + de^T y^k) - 2hg(x^*) + (\theta^k - \theta^*)^T \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} (\theta^k - \theta^*)$$

whose expectation decreases at appropriate rate. Here we denote

$$e = \begin{pmatrix} I \\ \vdots \\ I \end{pmatrix} \in \mathbb{R}^{np\times p}, \quad \theta^k = \begin{pmatrix} y_1^k \\ \vdots \\ y_n^k \\ x^k \end{pmatrix} = \begin{pmatrix} y^k \\ x^k \end{pmatrix} \in \mathbb{R}^{(n+1)p}, \quad \theta^* = \begin{pmatrix} \nabla f_1(x^*) \\ \vdots \\ \nabla f_n(x^*) \\ x^* \end{pmatrix} \in \mathbb{R}^{(n+1)p}$$

$$A = a_1 ee^T + a_2 I, B = be, C = cI$$

$\mathcal{L}(\theta^k)$ has parameters $\{a_1, a_2, b, c, d, h\}$.

Proof Outline

- Show that for appropriate $\sigma \geq 0$ and $\gamma \geq 0$ that

$$\text{(a) } \mathbb{E}(\mathcal{L}(\theta^k)|\mathcal{F}_{k-1}) \leq (1-\delta)\mathcal{L}(\theta^{k-1}),$$
$$\text{(b) } \mathcal{L}(\theta^k) \geq \gamma[g(x^k) - g(x^*)]$$

  where $\mathcal{F}_k$ is the σ-field of information from time 1 to k, that is, the σ-field generated by $i_1, ..., i_k$.

- Find parameters $\{a_1, a_2, b, c, d, h, \alpha, \gamma, \sigma\}$ that satisfies the above property. Using a SOCP solver that solves the parameter constraint, we have

$$\delta = \min(\frac{1}{8n}, \frac{\mu}{16L}), \gamma = 1$$

- Take expectation on both results, and combine them to have

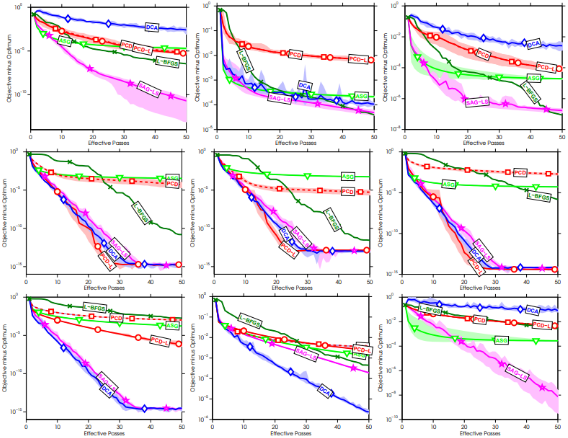
$$\mathbb{E}[g(x^k)] - g(x^*) \leq \left(1 - \min\{\frac{\mu}{16L}, \frac{1}{8n}\}\right)^k \mathcal{L}(\theta^0)$$

Proof Outline

- A slight modification of the above process gives the desired bound for general convex case $\mu = 0$.
- Plugging in determined parameters to get initial values of the Lyapunov function.

Comparison with Other Methods

- Binary classification using logistic regression, applied to 9 datasets (quantum, protein, covertype, rcv1, news, spam, rcv1Full, sido, alpha) compared against full-gradient or stochastic-gradient methods (AFG, L-BFGS, SG, ASG, LAG)
- SAG optimizes fast!



Effect of Non-Uniform Sampling

- Sampling in proportion to gradient's Lipschitz constants performs well.
- Intuition? We may not need to sample functions whose gradient changes slowly as much as ones whose gradient changes more quickly.
- Results for datasets where SAG didn't perform well.

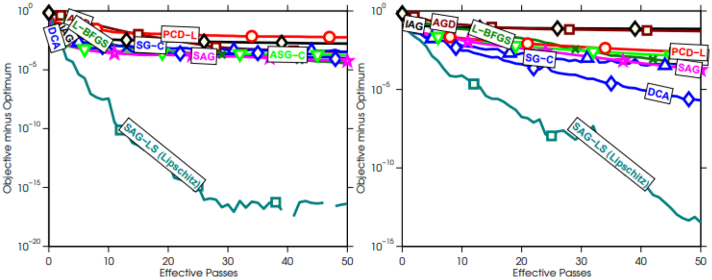


Table of Contents

SGD

Consider optimization problem

$$\min f(w) = \frac{1}{N} \sum_{i=1}^{N} f_i(w)$$

where $f_1, \ldots, f_N : \mathbb{R}^d \to \mathbb{R}$

SGD draw i_t randomly from $\{1, \cdots, N\}$ and perform

$$w_{t+1} = w_t - \eta_t \nabla f_{i_t}(w_t)$$

or more generally

$$w_{t+1} = w_t - \eta_t g_t(w_t, \xi_t)$$

where ξ_t is a random variable and $\mathbb{E}_{\xi_t}[g_t(w_t, \xi_t)] = \nabla f(w_t)$; *i.e.*, the expectation $\mathbb{E}[w_{t+1} | w_t]$ is identical to GD

Pros

- Each step only relies on a single derivative $\nabla f_i(\cdot)$, thus the computational cost is $\frac{1}{N}$ that of the GD
- Popular for large scale optimization

Cons

- The randomness introduces variance
- If $\|g_t(w_t, \xi_t)\|$ is large, then it has a relatively large variance which slows down the convergence

Convergence results

Suppose $f_i(w)$'s are β-smooth and convex; and $f(w)$ is α-strongly convex

GD

- As we choose $\eta_t < \frac{1}{\beta}$, we have **linear convergence** rate of $\mathcal{O}((1 - \frac{\alpha}{\beta})^t)$

SGD

- Due to the variance of random sampling, generally need to choose $\eta_t \sim \mathcal{O}(\frac{1}{t})$
- Then obtain a slower **sub-linear convergence** rate of $\mathcal{O}(\frac{1}{t})$

Motivation

The above implies we have a **trade-off**

- Slow computation per iteration and fast convergence for GD
- Fast computation per iteration and slow convergence for SGD

Then how can we improve the SGD?

- One practical issue for SGD : the **learning rate** η_t has to **decay to zero** - leads to slower convergence
- Need : allows us to use a **larger** learning rate η_t

Why do we have to use **small** learning rate?

- Due to the variance

Previous work – SAG (Schmidt et al. 2017)

Draw i_t randomly from $\{1, \cdots, N\}$ and

$$w_{t+1} = w_t - \frac{\eta_t}{N} \sum_{i=1}^{N} g_{i,t}$$

where

$$g_{i,t} = \begin{cases} \nabla f_i(w_t) & \text{if } i = i_t \\ g_{i,t-1} & \text{else} \end{cases}$$

- only set $g_{i_t,t} = \nabla f_{i_t}(w_t)$ for a randomly chosen i_t and all other $g_{i \neq i_t,t}$ are kept at their previous value
- can think of SAG as having a **memory**

$$\begin{bmatrix} — & g_{1,t} & — \\ — & g_{2,t} & — \\ & \vdots & \\ — & g_{N,t} & — \end{bmatrix}$$

Previous work – SDCA (Shalev-Shwartz and Zhang 2013)

SDCA applies randomized coordinate ascent to the dual of ridge regularized problems, and effective primal updates are similar to SAG

Consider the following problem with convex $\phi_i(w)$

$$w^* = \arg\min_w f(w), \qquad f(w) = \frac{1}{N}\sum_{i=1}^{N}\phi_i(w^T x_i) + \frac{\lambda}{2}\|w\|_2^2$$

The dual problem is

$$\max_{\alpha\in\mathbb{R}^d} D(\alpha) \text{ where } D(\alpha) = \left[\frac{1}{N}\sum_{i=1}^{N} -\phi_i^*\left(-\alpha_i\right) - \frac{\lambda}{2}\left\|\frac{1}{\lambda N}\sum_{i=1}^{N}\alpha_i x_i\right\|^2\right]$$

Turn it into $f_i(w) = \phi_i(x_i) + \frac{\lambda}{2}\|w\|_2^2$, α^* be a optimal solution of dual.

And define $w(\alpha_t) = \frac{1}{\lambda N}\sum_{i=1}^{N} \alpha_{i,t}$

- It is also known that $f(w^*) = D(\alpha^*)$
- And also we have $f(w) \geq D(\alpha)$
- The duality gap $f(w(\alpha_t)) - D(\alpha_t)$ is lower bounded by $f(w(\alpha_t)) - f(w^*)$

Stochastic Dual Coordinate Ascent rule, draw i_t randomly from $\{1, \cdots, N\}$

$$\alpha_{i,t+1} = \begin{cases} \alpha_{i,t} - \eta_t \left(\nabla \phi_i \left(w_t\right) + \lambda N \alpha_{i,t}\right) & i = i_t \\ \alpha_{i,t} & i \neq i_t \end{cases}$$

and then update w as $w_{t+1} = w_t + (\alpha_{t+1} - \alpha_t)$

Taking expectation yields the gradient descent rule

$$\mathbb{E} w_{t+1} | w_t = w_t - \eta_t \nabla f(w_t)$$

We can think of SDCA also as having a **memory**

$$\begin{bmatrix} — & \alpha_{1,t} & — \\ — & \alpha_{2,t} & — \\ & \vdots & \\ — & \alpha_{N,t} & — \end{bmatrix}$$

Both proposals **require storage** of all gradients (or dual variables), makes it unsuitable for more complex applications

SVRG (Johnson and Zhang 2013)

Keep a snapshot of $\tilde{w}$ every m SGD iterations while maintating the average gradient

$$\tilde{\mu} = \nabla f(\tilde{w}) = \frac{1}{N} \sum_{i=1}^{n} \nabla f_i(\tilde{w})$$

Update the parameter as the following rule

$$w_{t+1} = w_t - \eta_t (\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}) + \tilde{\mu})$$

SVRG is special case of SGD

$$g_t(w_t, \xi_t) = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}) + \tilde{\mu}$$

equivalently, SVRG is a SGD of the auxiliary function

$$\tilde{f}_{i_t}(w) := f_{i_t}(w) - (\nabla f_{i_t}(\tilde{w}) - \tilde{\mu})^T w$$

Since $\sum_{i=1}^{N}(\nabla f_i(\tilde{w}) - \tilde{\mu}) = 0$,

$$f(w) = \sum_{i=1}^{N} f_i(w) = \sum_{i=1}^{N} \tilde{f}_i(w)$$

Algorithm

Parameters: update frequency m and learning rate η
Initialize: $\tilde{w}_0$
for $s = 0, 1, \cdots$ **do**
  $\tilde{w} \leftarrow \tilde{w}_s$
  $\tilde{\mu} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\tilde{w})$
  $w_0 \leftarrow \tilde{w}$
  for $t = 0, 1, \cdots, m-1$ **do**
    Randomly pick $i_t \in \{1, 2, \cdots, N\}$ and update weight
    $w_{t+1} = w_t - \eta(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}) + \tilde{\mu})$
  end
  option 1: $\tilde{w}_{s+1} \leftarrow w_m$
  option 2: $\tilde{w}_{s+1} \leftarrow w_t$ for randomly chosen $t \in \{0, \cdots, m-1\}$
end

Algorithm 1: SVRG

Note

- When both $\tilde{w}$ and w_t converges to the same parameter w^*, then $\tilde{\mu} \to 0$
- If $\nabla f_i(\tilde{w}) \to \nabla f_i(w^*)$,then

$$\nabla f_i\left(w_t\right) - \nabla f_i(\tilde{w}) + \tilde{\mu} \quad \to \quad \nabla f_i\left(w_t\right) - \nabla f_i\left(w^*\right) \quad \to \quad 0$$

- Unlike SGD, the learning rate η_t for SVRG **does not have to decay**, which leads to **faster convergence** as one can use a relatively large learning rate.

Computational cost

- Each stage s requires $N + 2m$ gradient computations
- One may save the intermediate gradients and thus only $N + m$ gradient computations are needed
- It is natural to choose m to be the same order of N but slightly larger
  1. (for example) $m = 2N$ for convex problems
  2. $m = 5N$ for nonconvex problems

Convergence analysis

Theorem

Assume

- *For each* $f_i(w)$ *is* β*-smooth, convex and* $f(w)$ *is* α*-strongly convex*
- *SVRG with option 2,* $w^* := \arg\min_w f(w), R_0 := f(\tilde{w}_0) - f(w^*)$
- *m is sufficiently large so that*

$$\rho := \frac{1}{\alpha\eta(1-2\beta\eta)m} + \frac{2\beta\eta}{1-2\beta\eta} < 1$$

then

$$\mathbb{E}f(\tilde{w}_s) - f(w^*) \leq R_0\rho^{\mathbf{s}}$$

Proof

Given any i, consider $g_i(w) := f_i(w) - f_i(w^*) - \nabla f_i(w^*)^\top (w - w^*)$
Since $\nabla g_i(w^*) = 0$, $g_i(w^*) = \min_w g_i(w)$. Therefore,

$$0 = g_i(w^*) \leq \min_\eta [g_i(w - \eta \nabla g_i(w))]$$
$$\leq \min_\eta \left[g_i(w) - \eta \|\nabla g_i(w)\|_2^2 + \frac{\beta \eta^2}{2} \|\nabla g_i(w)\|_2^2 \right] = g_i(w) - \frac{1}{2\beta} \|\nabla g_i(w)\|_2^2$$

which implies,

$$\|\nabla f_i(w) - \nabla f_i(w^*)\|_2^2 \leq 2\beta \left[f_i(w) - f_i(w^*) - \nabla f_i(w^*)^\top (w - w^*) \right] \quad (1)$$

By summing (1) over $i = 1, \cdots, N$ and using the fact that $\nabla f(w^*) = 0$, we obtain

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla f_i(w) - \nabla f_i(w^*)\|_2^2 \leq 2\beta \left[f(w) - f(w^*)\right] \tag{2}$$

On the other hand, let $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}) + \tilde{\mu}$, then the conditional expectation w.r.t i_t conditioned on w_t is

$$\begin{aligned}
\mathbb{E}\|v_t\|_2^2 &\leq 2\mathbb{E}\left\|\nabla f_{i_t}(w_t) - \nabla f_{i_t}(w^*)\right\|_2^2 + 2\mathbb{E}\left\|[\nabla f_{i_t}(\tilde{w}) - \nabla f_{i_t}(w^*)] - \nabla f(\tilde{w})\right\|_2^2 \\
&= 2\mathbb{E}\left\|\nabla f_{i_t}(w_t) - \nabla f_{i_t}(w^*)\right\|_2^2 + 2\mathbb{E}\| \left[\nabla f_{i_t}(\tilde{w}) - \nabla f_{i_t}(w^*)\right] \\
&\qquad - \mathbb{E}\left[\nabla f_{i_t}(\tilde{w}) - \nabla f_{i_t}(w^*)\right] \|_2^2 \\
&\leq 2\mathbb{E}\left\|\nabla f_{i_t}(w_t) - \nabla f_{i_t}(w^*)\right\|_2^2 + 2\mathbb{E}\left\|\nabla f_{i_t}(\tilde{w}) - \nabla f_{i_t}(w^*)\right\|_2^2 \qquad (\because (2)) \\
&\leq 4\beta\left[f(w_t) - f(w^*) + f(\tilde{w}) - f(w^*)\right] \qquad (3)
\end{aligned}$$

This leads to,

$$
\begin{aligned}
\mathbb{E}\left\|w_{t+1}-w^*\right\|_2^2 &= \left\|w_t-w^*\right\|_2^2 - 2\eta\left(w_t-w^*\right)^\top \mathbb{E}[v_t] + \eta^2\mathbb{E}\left\|v_t\right\|_2^2 \\
&\leq \left\|w_t-w^*\right\|_2^2 - 2\eta\left(w_t-w^*\right)^\top \nabla f\left(w_t\right) \\
&\quad + 4\beta\eta^2\left[f\left(w_t\right)-f\left(w^*\right)+f(\tilde{w})-f\left(w^*\right)\right] \qquad (\because \text{(3)}, \mathbb{E}[v_t]=\nabla f(w_t)) \\
&\leq \left\|w_t-w^*\right\|_2^2 - 2\eta\left[f\left(w_t\right)-f\left(w^*\right)\right] \\
&\quad + 4\beta\eta^2\left[f\left(w_t\right)-f\left(w^*\right)+f(\tilde{w})-f\left(w^*\right)\right] \qquad (\because \text{convexity of } f(w)) \\
&= \left\|w_t-w^*\right\|_2^2 - 2\eta(1-2\beta\eta)\left[f\left(w_t\right)-f\left(w^*\right)\right] + 4\beta\eta^2\left[f(\tilde{w})-f\left(w^*\right)\right]
\end{aligned} \tag{4}
$$

We consider a fixed stage $s-1$, $\tilde{w}=\tilde{w}_{s-1}$ and $\tilde{w}_s$ is selected by option 2, then summing (4) over $t=0,\cdots,m-1$ and taking expectation,

$$\begin{aligned}
&\mathbb{E}\left\|w_m-w^*\right\|_2^2+2\eta(1-2\beta\eta)m\mathbb{E}\left[f\left(\tilde{w}_s\right)-f\left(w^*\right)\right] \\
&\qquad\leq \mathbb{E}\left\|w_0-w^*\right\|_2^2+4\beta m\eta^2\mathbb{E}\left[f(\tilde{w})-f\left(w^*\right)\right] \\
&\qquad= \mathbb{E}\left\|\tilde{w}-w^*\right\|_2^2+4\beta m\eta^2\mathbb{E}\left[f(\tilde{w})-f\left(w^*\right)\right] \\
&\qquad\leq \frac{2}{\alpha}\mathbb{E}\left[f(\tilde{w})-f\left(w^*\right)\right]+4\beta m\eta^2\mathbb{E}\left[f(\tilde{w})-f\left(w^*\right)\right] \\
&\qquad\qquad\qquad(\because \text{ strongly convexity of } f(w)) \\
&\qquad= 2\left(\alpha^{-1}+2\beta m\eta^2\right)\mathbb{E}\left[f(\tilde{w})-f\left(w^*\right)\right]
\end{aligned}$$

Thus we obtain

$$\mathbb{E}\left[f\left(\tilde{w}_s\right)-f\left(w^*\right)\right] \leq \left[\frac{1}{\alpha\eta(1-2\beta\eta)m}+\frac{2\beta\eta}{1-2\beta\eta}\right]\mathbb{E}\left[f\left(\tilde{w}_{s-1}\right)-f\left(w^*\right)\right]$$

which implies the desired bound $\mathbb{E}f(\tilde{w}_s)-f(w^*)\leq R_0\rho^{\mathbf{s}}$ □

Analysis

smooth but not strongly convex case

- A convergence rate of $\mathcal{O}(\frac{1}{t})$ may be obtained
- which improves the standard SGD convergence rate of $\mathcal{O}(\frac{1}{\sqrt{t}})$

Analysis

SDCA as Variance Reduction

It can be shown that both SDCA is connected to SVRG in the sense they are also a variance reduction methods for SGD

- The advantage of SDCA is that we may take a **larger step** when $t \to \infty$
  - Since $f(w^*) = D(\alpha^*),\ (w(\alpha_t), \alpha_t) \to (w^*, \alpha^*) \implies \nabla\phi_(w) + \lambda N\alpha \to 0$
  - It means that even if η_t stays bounded away from zero, the procedure can converge

SDCA is also a **variance reduction method** for SGD, which is similar to SVRG
But SVRG is is simpler, more intuitive, and easier to analyze

Experiments

Compare to SGD and SDCA with linear predictors(convex) and neural nets(nonconvex)

- The x-axis is computational cost measured by the number of gradient computations divided by N
- For SGD, it is the number of passes to go through the training data
- The interval m was set to 2N (convex) and 5N (nonconvex)
- The weights for SVRG were initialized by performing 1 iteration(convex) or 10 iterations(nonconvex) of SGD

Experiments

L2-regularized multiclass logistic regression (convex optimization) on MNIST

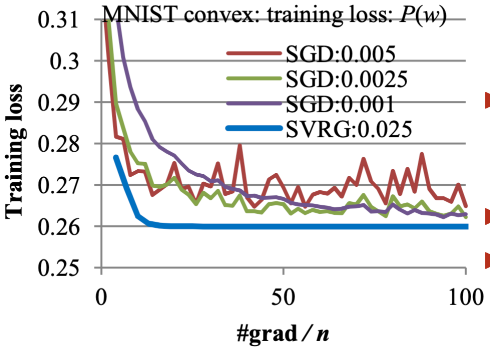


Figure: Training loss comparison with SGD with fixed LR

- When a relatively large learning rate η is used with SGD, it oscillates above the minimum and never goes down to the minimum
- SVRG smoothly goes down faster than SGD
- Relatively large η with SVRG leads to faster convergence

Experiments

L2-regularized multiclass logistic regression (convex optimization) on MNIST

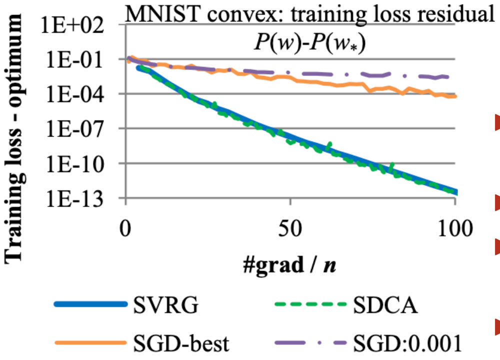


Figure: Training loss residual $f(w) - f(w^*)$; comparison with best-tuned SGD and SDCA

- SGD with best scheduling of exponential decay, adaptive
- SVRG's loss residual goes down exponentially
- SVRG is competitive with SDCA (the two lines are almost overlapping)
- SVRG decreases faster than SGD-best

Experiments

L2-regularized multiclass logistic regression (convex optimization) on MNIST

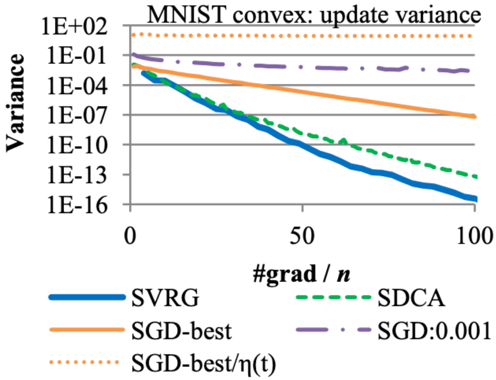


Figure: Variance of weight update

- Including multiplication with the learning rate
- SGD with a fixed learning rate (‘SGD:0.001’) stays high
- The variance of the best-tuned SGD decreases
- SVRG decreases faster than SGD-best

Experiments

Neural nets on MNIST

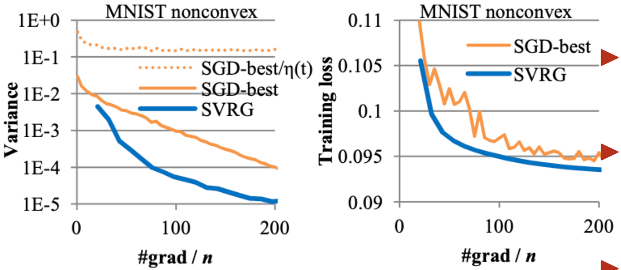


Figure: Neural net results(nonconvex)

- FC layer of 100 nodes and softmax output; sigmoid activation and L2 regularization
- Mini-batches of size 10
- SVRG reduces the variance and converges faster than the best-tuned SGD
- SDCA and SAG are not practical for neural nets due to their memory requirement

Conclusion

- Introduces an explicit variance reduction method for SGD
- Provide that this method enjoys the same fast convergence rate as those of SDCA and SAG
- unlike SDCA or SAG, this method does not require the storage of gradients, and thus is more easily applicable to complex problems

Any questions?

References I

Johnson, Rie and Tong Zhang (2013). "Accelerating stochastic gradient descent using predictive variance reduction". In: *Advances in neural information processing systems.*

Schmidt, Mark, Nicolas Le Roux, and Francis Bach (2017). "Minimizing finite sums with the stochastic average gradient". In: *Mathematical Programming.*

Shalev-Shwartz, Shai and Tong Zhang (2013). "Stochastic dual coordinate ascent methods for regularized loss minimization". In: *JMLR.*