Sub-First-Order Methods

Stephen J. Wright[1]

[2]Computer Sciences Department,
University of Wisconsin-Madison.

IMA, August 2016

Stochastic Gradient Methods

Deal with (weakly or strongly) convex f.

- Allow f nonsmooth.
- Can't get function values $f(x)$ easily.
- At any feasible x, have access only to a cheap unbiased estimate of an element of the subgradient ∂f.

Common settings are:

$$f(x) = E_\xi F(x, \xi),$$

where ξ is a random vector with distribution P over a set Ξ. Special case:

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x),$$

where each f_i is convex and nonsmooth.
(We focus on this finite-sum formulation, but the ideas generalize.)

Applications

This setting is useful for machine learning formulations. Given data $x_i \in \mathbb{R}^n$ and labels $y_i = \pm 1$, $i = 1, 2, \ldots, m$, find w that minimizes

$$\tau\psi(w) + \frac{1}{m}\sum_{i=1}^{m} \ell(w; x_i, y_i),$$

where ψ is a regularizer, $\tau > 0$ is a parameter, and ℓ is a loss. For linear classifiers/regressors, have the specific form $\ell(w^T x_i, y_i)$.

Example: SVM with hinge loss $\ell(w^T x_i, y_i) = \max(1 - y_i(w^T x_i), 0)$ and $\psi = \|\cdot\|_1$ or $\psi = \|\cdot\|_2^2$.

Example: Logistic regression: $\ell(w^T x_i, y_i) = \log(1 + \exp(y_i w^T x_i))$. In regularized version may have $\psi(w) = \|w\|_1$.

Example: Deep Learning (the killer app!).

Subgradients

Recall: For each x in domain of f, g is a *subgradient of f at x* if

$$f(z) \geq f(x) + g^T(z - x), \qquad \text{for all } z \in \text{dom } f.$$

- Right-hand side is a *supporting hyperplane*.
- The set of subgradients is called the *subdifferential*, denoted by $\partial f(x)$.
- When f is differentiable at x, have $\partial f(x) = \{\nabla f(x)\}$.

We have strong convexity with modulus $m > 0$ if

$$f(z) \geq f(x) + g^T(z-x) + \frac{1}{2} m \|z-x\|^2, \quad \text{for all } x, z \in \text{dom } f \text{ with } g \in \partial f(x).$$

Generalizes the assumption $\nabla^2 f(x) \succeq mI$ made earlier for smooth functions.

Classical Stochastic Gradient

For the finite-sum objective, get a cheap unbiased estimate of the gradient $\nabla f(x)$ by choosing an index $i \in \{1, 2, \ldots, m\}$ uniformly at random, and using $\nabla f_i(x)$ to estimate $\nabla f(x)$.

Basic SA Scheme: At iteration k, choose i_k i.i.d. uniformly at random from $\{1, 2, \ldots, m\}$, choose some $\alpha_k > 0$, and set

$$x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k).$$

Note that x_{k+1} depends on all random indices up to iteration k, i.e. $i_{[k]} := \{i_1, i_2, \ldots, i_k\}$.

When f is strongly convex, the analysis of convergence of expected square error $E(\|x_k - x^*\|^2)$ is fairly elementary — see Nemirovski et al (2009).

Define $a_k = \frac{1}{2} E(\|x_k - x^*\|^2)$. Assume there is $M > 0$ such that

$$\frac{1}{m} \sum_{i=1}^{m} \|\nabla f_i(x)\|_2^2 \leq M.$$

Rate: $1/k$

Thus

$$\begin{aligned}
&\frac{1}{2}\|x_{k+1} - x^*\|_2^2 \\
&\qquad = \frac{1}{2}\|x_k - \alpha_k \nabla f_{i_k}(x_k) - x^*\|^2 \\
&\qquad = \frac{1}{2}\|x_k - x^*\|_2^2 - \alpha_k (x_k - x^*)^T \nabla f_{i_k}(x_k) + \frac{1}{2}\alpha_k^2 \|\nabla f_{i_k}(x_k)\|^2.
\end{aligned}$$

Taking expectations, get

$$a_{k+1} \leq a_k - \alpha_k E[(x_k - x^*)^T \nabla f_{i_k}(x_k)] + \frac{1}{2}\alpha_k^2 M^2.$$

For middle term, have

$$\begin{aligned}
E[(x_k - x^*)^T \nabla f_{i_k}(x_k)] &= E_{i_{[k-1]}} E_{i_k}[(x_k - x^*)^T \nabla f_{i_k}(x_k) | i_{[k-1]}] \\
&= E_{i_{[k-1]}} (x_k - x^*)^T g_k,
\end{aligned}$$

... where

$$g_k := E_{i_k}[\nabla f_{i_k}(x_k)|i_{[k-1]}] \in \partial f(x_k).$$

By strong convexity, have

$$(x_k - x^*)^T g_k \geq f(x_k) - f(x^*) + \frac{1}{2} m\|x_k - x^*\|^2 \geq m\|x_k - x^*\|^2.$$

Hence by taking expectations, we get $E[(x_k - x^*)^T g_k] \geq 2ma_k$. Then, substituting above, we obtain

$$a_{k+1} \leq (1 - 2m\alpha_k)a_k + \frac{1}{2}\alpha_k^2 M^2.$$

When

$$\alpha_k \equiv \frac{1}{km},$$

a neat inductive argument (below) reveals the $1/k$ rate:

$$a_k \leq \frac{Q}{2k}, \qquad \text{for } Q := \max\left(\|x_1 - x^*\|^2, \frac{M^2}{m^2}\right).$$

Inductive Proof of $1/k$ Rate

Clearly true for $k = 1$. Otherwise:

$$\begin{aligned}
a_{k+1} &\leq (1 - 2m\alpha_k)a_k + \frac{1}{2}\alpha_k^2 M^2 \\
&\leq \left(1 - \frac{2}{k}\right) a_k + \frac{M^2}{2k^2 m^2} \\
&\leq \left(1 - \frac{2}{k}\right) \frac{Q}{2k} + \frac{Q}{2k^2} \\
&= \frac{(k-1)}{2k^2} Q \\
&= \frac{k^2 - 1}{k^2} \frac{Q}{2(k+1)} \\
&\leq \frac{Q}{2(k+1)},
\end{aligned}$$

as claimed.

But... What if we don't know m? Or if $m = 0$?

The choice $\alpha_k = 1/(km)$ requires strong convexity, with knowledge of the modulus m. An underestimate of m can greatly degrade the performance of the method (see example in Nemirovski et al. 2009).

Now describe a *Robust Stochastic Approximation* approach, which has a rate $1/\sqrt{k}$ (in function value convergence), and works for weakly convex nonsmooth functions and is not sensitive to choice of parameters in the step length.

This is the approach that generalizes to *mirror descent*, as discussed later.

Robust SA

At iteration k:

- set $x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k)$ as before;
- set

$$\bar{x}_k = \frac{\sum_{i=1}^k \alpha_i x_i}{\sum_{i=1}^k \alpha_i}.$$

For any $\theta > 0$, choose step lengths to be

$$\alpha_k = \frac{\theta}{M\sqrt{k}}.$$

Then $f(\bar{x}_k)$ converges to $f(x^*)$ in expectation with rate approximately $(\log k)/k^{1/2}$.

(The choice of θ is not critical.)

Analysis of Robust SA

The analysis is again elementary. As above (using i instead of k), have:

$$\alpha_i E[(x_i - x^*)^T g_i] \leq a_i - a_{i+1} + \frac{1}{2}\alpha_i^2 M^2.$$

By convexity of f, and $g_i \in \partial f(x_i)$:

$$f(x^*) \geq f(x_i) + g_i^T(x^* - x_i),$$

thus

$$\alpha_i E[f(x_i) - f(x^*)] \leq a_i - a_{i+1} + \frac{1}{2}\alpha_i^2 M^2,$$

so by summing iterates $i = 1, 2, \ldots, k$, telescoping, and using $a_{k+1} > 0$:

$$\sum_{i=1}^{k} \alpha_i E[f(x_i) - f(x^*)] \leq a_1 + \frac{1}{2} M^2 \sum_{i=1}^{k} \alpha_i^2.$$

Thus dividing by $\sum_{i=1} \alpha_i$:

$$E\left[\frac{\sum_{i=1}^k \alpha_i f(x_i)}{\sum_{i=1}^k \alpha_i} - f(x^*)\right] \leq \frac{a_1 + \frac{1}{2}M^2 \sum_{i=1}^k \alpha_i^2}{\sum_{i=1}^k \alpha_i}.$$

By convexity, we have

$$f(\bar{x}_k) = f\left(\frac{\sum_{i=1}^k \alpha_i x_i}{\sum_{i=1}^k \alpha_i}\right) \leq \frac{\sum_{i=1}^k \alpha_i f(x_i)}{\sum_{i=1}^k \alpha_i},$$

so obtain the fundamental bound:

$$E[f(\bar{x}_k) - f(x^*)] \leq \frac{a_1 + \frac{1}{2}M^2 \sum_{i=1}^k \alpha_i^2}{\sum_{i=1}^k \alpha_i}.$$

By substituting $\alpha_i = \frac{\theta}{M\sqrt{i}}$, we obtain

$$
\begin{aligned}
E[f(\bar{x}_k) - f(x^*)] &\leq \frac{a_1 + \frac{1}{2}\theta^2 \sum_{i=1}^{k} \frac{1}{i}}{\frac{\theta}{M} \sum_{i=1}^{k} \frac{1}{\sqrt{i}}} \\
&\leq \frac{a_1 + \theta^2 \log(k+1)}{\frac{\theta}{M}\sqrt{k}} \\
&= M\left[\frac{a_1}{\theta} + \theta \log(k+1)\right] \frac{1}{\sqrt{k}}.
\end{aligned}
$$

That's it!

There are other variants — periodic restarting, averaging just over the recent iterates. These can be analyzed with the basic bound above.

Constant Step Size

We can also get rates of approximately $1/k$ for the strongly convex case, *without* performing iterate averaging. The tricks are to

- define the desired threshold ϵ for a_k in advance, and
- use a constant step size.

Recall the bound on a_{k+1} from a few slides back, and set $\alpha_k \equiv \alpha$:

$$a_{k+1} \leq (1 - 2m\alpha)a_k + \frac{1}{2}\alpha^2 M^2.$$

Apply this recursively to get

$$a_k \leq (1 - 2m\alpha)^k a_0 + \frac{\alpha M^2}{4m}.$$

Given $\epsilon > 0$, find α and K so that both terms on the right-hand side are less than $\epsilon/2$. The right values are:

$$\alpha := \frac{2\epsilon\mu}{M^2}, \qquad K := \frac{M^2}{4\epsilon\mu^2} \log\left(\frac{a_0}{2\epsilon}\right).$$

Constant Step Size, continued

Clearly the choice of α guarantees that the second term is less than $\epsilon/2$.

For the first term, we obtain k from an elementary argument:

$$\begin{aligned}
& (1-2m\alpha)^k a_0 \leq \epsilon/2 \\
\Leftrightarrow \quad & k \log(1-2m\alpha) \leq -\log(2a_0/\epsilon) \\
\Leftarrow \quad & k(-2m\alpha) \leq -\log(2a_0/\epsilon) \qquad \text{since } \log(1+x) \leq x \\
\Leftrightarrow \quad & k \geq \frac{1}{2m\alpha} \log(2a_0/\epsilon),
\end{aligned}$$

from which the result follows, by substituting for α in the right-hand side.

If m is underestimated by a factor of β, we undervalue α by the same factor, and K increases by $1/\beta$. (Easy modification of the analysis above.)

Thus, underestimating m gives a mild performance penalty.

Constant Step Size: Summary

PRO: Avoid averaging, $1/k$ sublinear convergence, insensitive to underestimates of m.

CON: Need to estimate probably unknown quantities: besides m, we need M (to get α) and a_0 (to get K).

We use constant size size in the parallel SG approach HOGWILD!, *to be described later.*

But the step is chosen by trying different options and seeing which seems to be converging fastest. We don't actually try to estimate all the quantities in the theory and construct α that way.

Mirror Descent

The step from x_k to x_{k+1} can be viewed as the solution of a subproblem:

$$x_{k+1} = \arg\min_z \nabla f_{i_k}(x_k)^T(z - x_k) + \frac{1}{2\alpha_k}\|z - x_k\|_2^2,$$

a linear estimate of f plus a prox-term. This provides a route to handling constrained problems, regularized problems, alternative prox-functions.

For the constrained problem $\min_{x\in\Omega} f(x)$, simply add the restriction $z \in \Omega$ to the subproblem above.

We may use other prox-functions in place of $(1/2)\|z - x\|_2^2$ above. Such alternatives may be particularly well suited to particular constraint sets Ω.

Mirror Descent is the term used for such generalizations of the SA approaches above.

Mirror Descent cont'd

Given constraint set Ω, choose a norm $\|\cdot\|$ (not necessarily Euclidean). Define the *distance-generating function* ω to be a strongly convex function on Ω with modulus 1 with respect to $\|\cdot\|$, that is,

$$(\omega'(x) - \omega'(z))^T(x - z) \geq \|x - z\|^2, \quad \text{for all } x, z \in \Omega,$$

where $\omega'(\cdot)$ denotes an element of the subdifferential.

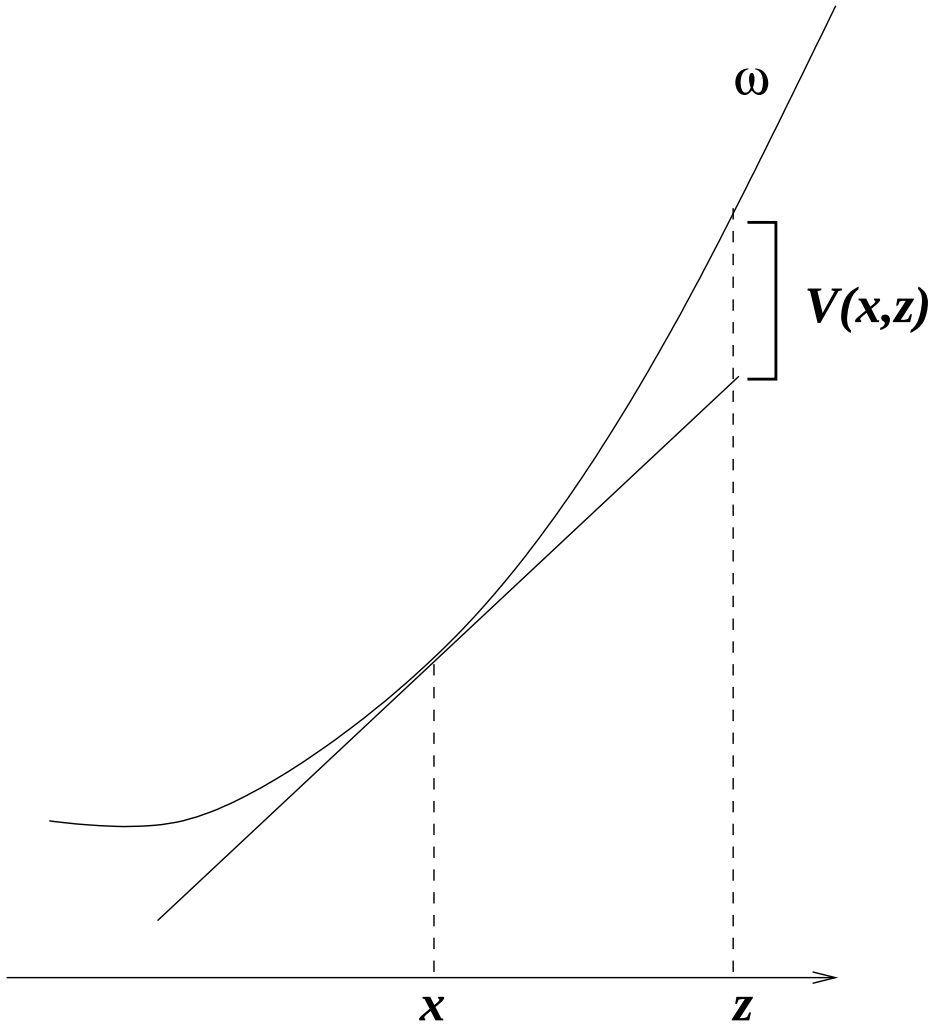
Now define the *prox-function* $V(x, z)$ as follows:

$$V(x, z) = \omega(z) - \omega(x) - \omega'(x)^T(z - x).$$

This is also known as the *Bregman distance*. We can use it in the subproblem in place of $\frac{1}{2}\|\cdot\|^2$:

$$x_{k+1} = \arg\min_{z \in \Omega} \nabla f_{i_k}(x_k)^T(z - x_k) + \frac{1}{\alpha_k} V(z, x_k).$$

Bregman distance is the deviation of ω from linearity:



Bregman Distances: Examples

For *any* Ω, we can use $\omega(x) := (1/2)\|x - \bar{x}\|_2^2$, leading to the "universal" prox-function

$$V(x, z) = (1/2)\|x - z\|_2^2$$

For the simplex

$$\Omega = \{x \in \mathbb{R}^n : x \geq 0, \sum_{i=1}^{n} x_i = 1\},$$

we can use instead the 1-norm $\|\cdot\|_1$, choose ω to be the entropy function

$$\omega(x) = \sum_{i=1}^{n} x_i \log x_i,$$

leading to Bregman distance (Kullback-Liebler divergence)

$$V(x, z) = \sum_{i=1}^{n} z_i \log(z_i/x_i),$$

which is standard measure of distance between two probability

Minibatching

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x),$$

can clump the f_i into disjoint "minibatches." Then write

$$f(x) = \frac{1}{b} \sum_{j=1}^{b} f_{[j]}(x),$$

where

$$f_{[j]}(x) = \sum_{i \in \mathcal{B}_j} f_i(x),$$

and $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_b$ is a partition of $\{1, 2, \ldots, n\}$.

Then apply SG to the batched form. Advantages:

- bigger chunks of work may make for more efficient implementation.
- lower variance in estimate of gradient $\nabla f(x)$ by $\nabla f_{[j]}(x)$.
- allows parallelism — the work of evaluating $\nabla f_i(x)$ for $i \in \mathcal{B}_j$ can be farmed out to various processors or cores.

Applications to SVM

SA techniques have an obvious application to linear SVM classification. In fact, they were proposed in this context and analyzed independently by researchers in the ML community for some years.

Codes: SGD (Bottou), PEGASOS (Shalev-Schwartz et al, 2007).

Tutorial: *Stochastic Optimization for Machine Learning*, Tutorial by N. Srebro and A. Tewari, ICML 2010 for many more details on the connections between stochastic optimization and machine learning.

Related Work: Zinkevich (ICML, 2003) on online convex programming. Aiming to approximate the minimize the average of a sequence of convex functions, presented sequentially. No i.i.d. assumption, regret-based analysis. Take steplengths of size $O(k^{-1/2})$ in gradient $\nabla f_k(x_k)$ of latest convex function. Average regret is $O(k^{-1/2})$.

Parallel Stochastic Gradient

Several approaches tried, for $f(x) = \sum_{i=1}^{m} f_i(x)$.

- **Asynchronous:** HOGWILD!**:** Each core grabs the centrally-stored x and evaluates $\nabla f_i(x)$ for some random i, then writes the updates back into x (Niu, Ré, Recht, Wright, NIPS, 2011).
- **Synchronous:** "Internal" parallelism in evaluation of gradient $\nabla f_{[j]}(x)$ for minibatch $\mathcal{B}_j$. (No changes needed to analysis.)

HOGWILD!: Each processor runs independently:

1. Sample i uniformly from $\{1, 2, \ldots, m\}$;
2. Read current state of x and evaluate $g_i = \nabla f_i(x)$;
3. Update $x \leftarrow x - \alpha g_i$;

HOGWILD! Convergence

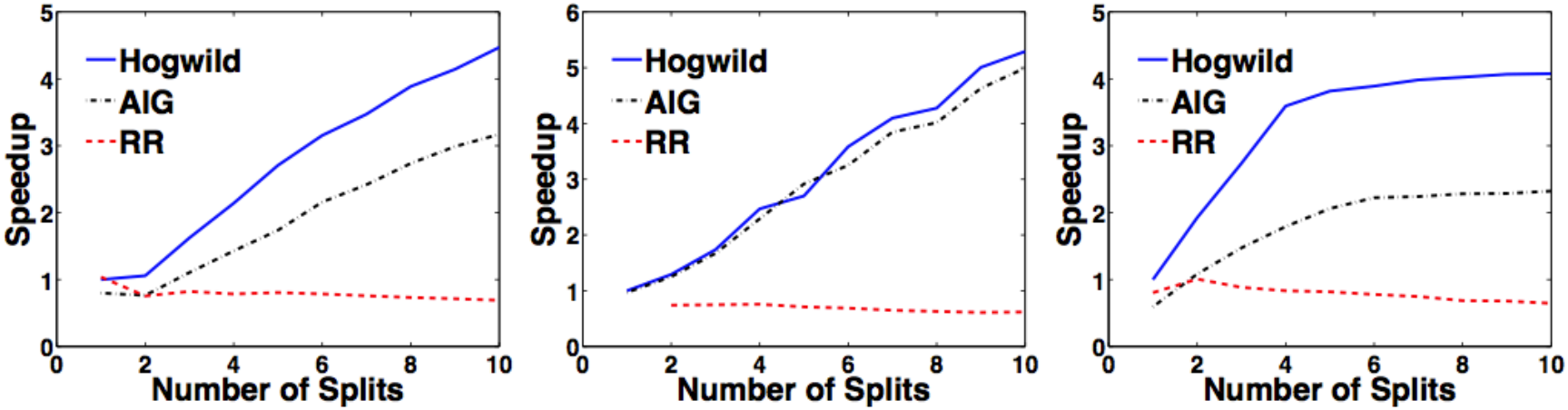
Analysis of asynchronous methods is nontrivial

- Updates can be old by the time they are applied, but we assume a bound τ on their age. This value τ is related to the number of cores involved in the computation.
- Processors can overwrite each other's work, but this effect is less important if each f_i depends on just a small number of components of x

Analysis of Niu et al (2011) simplified / generalized by Richtarik (2012) and many others.

Essentially show that similar $1/k$ rate to serial SG are possible provided that the maximum age τ of the updates is not too large.

HOGWILD! Performance



HOGWILD! compared with averaged gradient (AIG) and round-robin (RR). Experiments run on a 12-core machine. (10 cores used for gradient evaluations, 2 cores for data shuffling.)

HOGWILD! Performance

	data set	size (GB)	ρ	Δ	time (s)	speedup
SVM	RCV1	0.9	4.4E-01	1.0E+00	10	4.5
	Netflix	1.5	2.5E-03	2.3E-03	301	5.3
MC	KDD	3.9	3.0E-03	1.8E-03	878	5.2
	JUMBO	30	2.6E-07	1.4E-07	9,454	6.8
	DBLife	0.003	8.6E-03	4.3E-03	230	8.8
CUTS	Abdomen	18	9.2E-04	9.2E-04	1,181	4.1

Stochastic Gradient for Regularized Objectives

Suppose a regularized objective:

$$f(x) + \tau\psi(x) = \sum_{i=1}^{m} f_i(x) + \tau\psi(x),$$

with ψ convex, nonsmooth, simple, as usual.

Since convergence theory for SG applies to *nonsmooth* convex functions, can write this as

$$\frac{1}{m}\sum_{i=1}^{m} \left[f_i(x) + \tau\psi(x)\right],$$

and apply the previous algorithm and analysis. The subgradient estimate used at each step would be

$$\nabla f_i(x) + \tau h, \quad \text{for any } h \in \partial\psi(x),$$

for some randomly selected $i \in \{1, 2, \ldots, m\}$.

However this approach ignores the structure in ψ. When $\psi(x) = \|x\|_1$, it typically produces nonsparse solutions.

We noted earlier that for objective f, the step can be obtained from the subproblem

$$x_{k+1} = \arg\min_z \nabla f_{i_k}(x_k)^T(z - x_k) + \frac{1}{2\alpha_k}\|z - x_k\|_2^2.$$

A natural extension to the regularized case $f + \tau\psi$ would be:

$$x_{k+1} = \arg\min_z \nabla f_{i_k}(x_k)^T(z - x_k) + \frac{1}{2\alpha_k}\|z - x_k\|_2^2 + \tau\psi(z),$$

which can be solved via the prox operator for ψ. (FOBOS, COMID; Duchi and Singer, 2009)

Because of high variance in the gradient estimates ∇f_i, behavior of this approach can be erratic, with poor sparsity.

Example

For scalar $x \in \mathbb{R}$, define

$$f_i(x) = \left(4\frac{i-1}{m-1} - 2\right) x.$$

(Coefficient of x is in range $[-2, 2]$.) We have $(1/m)\sum_{i=1}^{m} f_i(x) = 0$ for all x! Define $\tau = 1$ and $\psi(x) = |x|$. Then the composite objective is

$$f(x) + \tau\psi(x) = |x|,$$

with minimizer $x^* = 0$. Suppose we start at the solution and take a prox-SG step, as above. The subproblem is

$$x_1 := \arg\min_z \left(4\frac{i_0 - 1}{m-1} - 2\right) z + |z| + \frac{1}{2\alpha_0} z^2.$$

- If i_0 is less than $m/4$, we have $x_1 > 0$;
- If i_0 is greater than $3m/4$, we have $x_1 < 0$;
- Otherwise, $x_1 = 0$.

With about 50% likelihood, we have $x_1 \neq 0$ — moves away from solution!

Does Averaging Help?

If we run classical SG for many iterates, arbitrarily many of them will be nonzero.

What if we take the weighted average of the *primal* iterates?

$$\bar{x}_k = \frac{\sum_{i=1}^k \alpha_i x_i}{\sum_{i=1}^k \alpha_i}.$$

This is almost certain to be nonzero, for *all* iterates.

But *dual averaging* may help! We average all gradients seen so far, and use this as the first-order terms in the subproblem.

Dual Averaging

(Nesterov, 2009) For min $f(x)$ with f convex, nonsmooth. Use subgradients $g_i \in \partial f(x_i)$ and average, to obtain

$$\bar{g}_k = \frac{1}{k} \sum_{i=1}^{k} g_i.$$

Step:

$$x_{k+1} := \min_x \bar{g}_k^T x + \frac{\gamma}{\sqrt{k}} \|x - x_0\|_2^2, \quad \text{for some constant } \gamma > 0.$$

- The last term is always centered at the *first* iterate x_0.
- Gradient information is averaged over all steps, with equal weights.
- γ is constant - results can be sensitive to this value.
- The approach still works for convex nondifferentiable f, where $\nabla f(x_i)$ is replaced by a vector from the subgradient $\partial f(x_i)$.

Dual Averaging: Convergence

Nesterov proves convergence for *averaged* iterates:

$$\bar{x}_{k+1} = \frac{1}{k+1} \sum_{i=0}^{k} x_i.$$

Provided the iterates and the solution x^* lie within some ball of radius D around x_0, we have

$$f(\bar{x}_{k+1}) - f(x^*) \leq \frac{C}{\sqrt{k}},$$

where C depends on D, a uniform bound on $\|\nabla f(x)\|$, and γ (coefficient of stabilizing term).

Note: There's averaging in both primal (x_i) and dual ($\nabla f(x_i)$) spaces.

Generalizes easily and robustly to the case in which only estimated gradients or subgradients are available.

Averaging smooths the errors in the individual gradient estimates.

Regularized Dual Averaging (RDA)

Dual averaging was extended to the regularized case by Xiao (2010):

$$\min \frac{1}{m} \sum_{i=1}^{m} f_i(x) + \tau\psi(x).$$

Subproblem introduces regularization term explicitly:

$$x_{k+1} = \arg\min_x \bar{g}_k^T x + \tau\psi(x) + \frac{\gamma}{\sqrt{k}}\|x - x_0\|^2.$$

Optimality conditions:

$$0 \in \bar{g}_k + \tau\partial\psi(x) + \frac{2\gamma}{\sqrt{k}}(x - x_0).$$

Optimality conditions for original problem:

$$0 \in \nabla f(^*) + \tau\partial\psi(x^*).$$

In the limit, have $\bar{g}_k \to \nabla f(x^*)$, $\gamma/\sqrt{k} \to 0$, so the subproblem is consistent with the original problem.

Manifold Identification in RDA

Under convexity assumptions, can show that $\bar{g}_k$ approaches $\nabla f(x^*)$ in expectation, with decreasing variance, at rate $k^{-1/4}$. (Faster if f is strongly convex).

Thus, under the usual assumptions of partial smoothness of ϕ_τ and nondegeneracy at x^*, a dense sequence of (non-averaged) iterates $\{x^k\}$ eventually stays on the optimal manifold $\mathcal{M}$, with probability $1 - O(k^{-1/4})$.

(The $O(\cdot)$ constant does not depend on problem dimension n.)

Motivates a 2-phase algorithm in which we switch to a different strategy (e.g. approximate reduced Newton method) on the low-dimensional manifold identified by RDA.

(Lee, Wright, 2011)

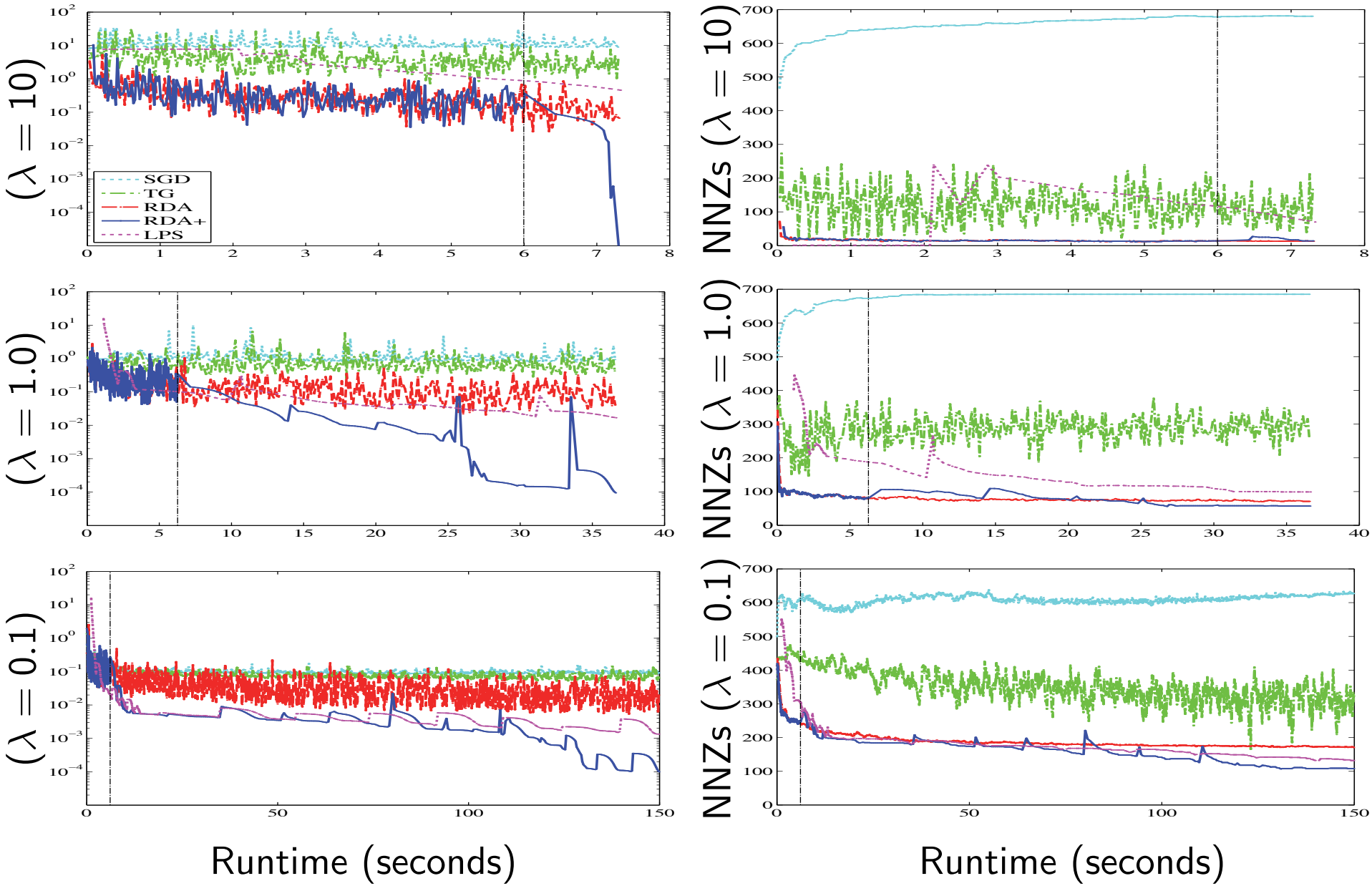
Computations

Next slide shows computations with MNIST data set (logistic regression for classifying digits 6 and 7).

- RED: Regularized Dual Averaging
- BLUE: Regularized Dual Averaging, with second phase
- GREEN: Truncated Gradient
- LIGHT BLUE: Stocastic Gradient
- PURPLE: Lasso-PatternSearch (a type of two-metric gradient projection) (Shi et al, 2008)

Vertical line indicates switch from RDA to reduced-space search.

- The second phase makes convergence faster (BLUE)
- RDA (and accelerated variant) gives sparser solutions.


($\lambda = 10$)
($\lambda = 1.0$)
($\lambda = 0.1$)
NNZs ($\lambda = 10$)
NNZs ($\lambda = 1.0$)
NNZs ($\lambda = 0.1$)
SGD
TG
RDA
RDA+
LPS
Runtime (seconds)
Runtime (seconds)

References I