Fast Stochastic Optimization Algorithms for ML

Aaditya Ramdas

April 20, 2015

This lecture is about efficient algorithms for *minimizing finite sums*

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w) \quad \text{or} \quad \min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w) + \frac{\lambda}{2} \|w\|^2$$

for known functions f_i and given λ. This is not necessarily what we want to minimize, but if we did, this lecture is relevant.

1 Brief Introduction

For linear *prediction* problems our aim is usually to minimize the true risk

$$\min_{w \in \mathbb{R}^d} \mathbb{E}_{(x,y) \sim P} L(w; x, y)$$

where P is a joint distribution on $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \mathbb{R}$ and w is a linear predictor, which is a vector in $\mathbb{R}^d$. More generally, we could minimize over functions f in some more complex nonlinear function class, like a Reproducing Kernel Hilbert Space.

For regression, y is real valued, L is the often squared loss (but not always; for example see Least Absolute Deviation regression) and w is the best linear fit to the data. For binary classification, y is binary, L is often the 0/1 loss and w is the best hyperplane separating the two sets of samples. Since the 0/1 loss is nonconvex, we often use convex upper bounds for the 0/1 loss - examples of such *surrogate* losses include the logistic, hinge, exponential and squared losses.

Since we don't know the true underlying P but only observe its realization through samples (data), we are often in a position where we choose to plug-in the empirical distribution P^n into the above expression, and instead minimize the empirical risk

$$\min_{w} F(w) := \frac{1}{n} \sum_{i=1}^{n} L(w; x_i, y_i)$$

To avoid overfitting, we use regularizers like $\|w\|^2$. In this lecture, the loss of w on the i-th data sample is represented as $f_i(w) := L(w; x_i, y_i)$ or $f_i(w) := L(w; x_i, y_i) + \frac{\lambda}{2}\|w\|^2$.

There are a huge variety of settings:

1. f_i could be differentiable (logistic) or nonsmooth (hinge) - in the latter case, subgradients are necessary.

2. If differentiable, f_i could have Lipschitz gradients.

3. f_i could be strongly convex (say if regularizer is used), or F could be strongly convex without each f_i being strongly convex (like OLS).

4. For fixed time budget T, we can measure progress by looking at point error $\|w_T - w^*\|^2$ or at function error $F(w_T) - F(w^*)$ or use $\bar{w}_T = (w_1 + ... + w_T)/T$ instead.

5. We may know T before hand, or may want the guarantee at *any* time point t (so the step size cannot use T since it is unknown).

6. We may want to know how many steps we need to get an pre-specified accuracy of ϵ for point error, or function error.

In this lecture, I will give talk more about the smooth and strongly convex setting. The associated references are more complete, but we can do a full course discussing these.

2 (Stochastic) Gradient Descent (SGD/GD)

Gradient descent is a deterministic algorithm, but it is very expensive. Even though it only needs $L/\lambda \log(1/\epsilon)$ steps in the smooth and strongly convex setting (for a function error of ϵ), it requires going through the whole dataset once at every iteration, taking nd time. Hence, we prefer *stochastic methods*, that only access one datapoint at every iteration.

SGD just picks a random index from 1 to n and updates $w_{t+1} = w_t - \eta \nabla f_i(w_t)$ where $\mathbb{E}\nabla f_i(w_t) = \nabla F(w_t)$. Note that this expectation is only over the randomness of uniformly choosing an index, not over the data. Step size η could be constant or a decreasing stepsize η_t (depending on if we know some problem dependent constants, or the time horizon).

Since it would take too much time to go into all problem variants, let it suffice to *loosely* say that stochastic gradient methods have a function error suboptimality of $O(1/T)$ in the strongly convex case, and of $O(1/\sqrt{T})$ in the convex setting.

There are lower bounds that prove that, in the black box setting for convex optimization, with only access to noisy gradients of some unknown function F, the aforementioned rates for minimizing F are essentially optimal in T (other quantities matter too, most importantly the dimension d, and the geometry of the set S if the minimization is being done over S).

So, are we done because we have matching lower bounds?

3 The suboptimality of SGD for minimizing finite sums

In 2012, in a game-changing paper, Schmidt, Le-Roux and Bach introduced an algorithm SAG that *beat* black box stochastic gradient methods, getting $O(1/T)$ rates for convex functions, and $O(\rho^T)$ rates for strongly convex functions for some $\rho < 1$.

The reason they were able to get around the lower bounds, is because

1. We are not interested in minimizing any arbitrary F, but one that is a finite sum (n is treated as fixed).
2. We are not restricted in choosing our stochastic gradient as $\nabla f_i(w_t)$, we can choose any g_t such that $\mathbb{E}g_t = \nabla F(w_t)$.
3. As a side note, we don't really need to have unbiased gradients, one can also tolerate (small and decreasing) biases. In fact, SAG updates are not unbiased (leading to much harder convergence proof).

Q: What is the point in choosing *different* unbiased estimators of the gradient? A: It's the variance that hurts the convergence rates!

What if we can (quickly) get estimates that are both (nearly) unbiased as well as variance that decreases to zero? We would approach the rates of gradient descent! How do we do this efficiently? A perfect method would be one which has

1. Provable rates for strongly convex functions as well as for convex (but not strongly convex) functions
2. Works for smooth as well as non-smooth functions.
3. Works even with an extra proximal regularization term.
4. Has a low storage (memory footprint) cost.
5. Is *adaptive* to strong convexity (the same algorithm is optimal in both settings with changing step-sizes or other parameters).
6. Has a simple proof.

4 SDCA, SVRG, SAG, SAGA, etc

Stochastic Dual Coordinate Ascent (Shai Shalev-Shwartz and Tong Zhang), Stochastic Variance-Reduced Gradient (Rie Johson and Tong Zhang), Stochastic Average Gradient Added-an-A-to-make-the-shortform-cool (Aaron Defazio, Francis Bach and Simon Lacoste-Julien) probably have the most relevant work - easy to Google. The last SAGA paper has a good introduction relating the different methods, and putting them in context. Recently, Gurbuzbalaban, Ozdaglar and Parrilo have a paper on a globally convergent incremental Newton method. Richtarik and coauthors have also extended this work. Let me give you a simple example of the SDCA algorithm.

This is for minimizing

$$\min_{w\in\mathbb{R}^d} F(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w) + \frac{\lambda}{2}\|w\|^2$$

where f_i is convex with L-smooth gradients.

SDCA. We choose a step-size $\eta < 1/\lambda n$, call $\beta := \eta\lambda n < 1$. Note that at optimality,

$$w^* = -\frac{1}{\lambda n}\sum_{i=1}^{n} \nabla f_i(w^*)$$

and what we will do is maintain "dual vectors" $\alpha_1, ..., \alpha_n$, so that

$$w^{(t)} = \frac{1}{\lambda n}\sum_{i=1}^{n} \alpha_i^{(t)}$$

The SDCA algorithm will make $w^{(t)} \to w^*$ and simultaneously $\alpha_i^{(t)} \to -\nabla f_i(w^*) =: \alpha_i^*$.

We initialize these dual vectors arbitrarily (and update $w^(0)$), and then at each time step, we uniformly randomly pick an index from 1 to n, and perform

$$\alpha_i^{(t)} = \alpha_i^{(t-1)} - \eta\lambda n(\nabla f_i(w^{(t-1)}) + \alpha_i^{(t-1)})$$

and hence correspondingly

$$w^{(t)} = w^{(t-1)} - \eta(\nabla f_i(w^{(t-1)}) + \alpha_i^{(t-1)}) := w^{(t-1)} - \eta g_t$$

That's it! Super-simple! Is this even an SGD-like algorithm? Yes, because

$$\mathbb{E} g_t = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w^{(t-1)}) + \frac{1}{n}\sum_{i=1}^{n} \alpha_i^{(t-1)} = \nabla F(w^{(t-1)})$$

because we always maintain $w^{(t-1)} = \frac{1}{\lambda n}\sum_{i=1}^{n} \alpha_i^{(t-1)}$.

Theorem:

$$\mathbb{E}\left[\frac{\lambda}{2}\|w^{(t)} - w^*\|^2 + \frac{1}{2Ln}\sum_{i=1}^{n}\|\alpha_i^{(t)} - \alpha^*\|^2\right] \le e^{-\eta\lambda t}\left[\frac{\lambda}{2}\|w^{(0)} - w^*\|^2 + \frac{1}{2Ln}\sum_{i=1}^{n}\|\alpha_i^{(0)} - \alpha^*\|^2\right]$$

For example, if $\eta = \frac{1}{L+\lambda n}$, then $\mathbb{E}[F(w^{(T)})] - F(w^*) \le \epsilon$ when $T = \Omega(\left(\frac{L}{\lambda} + n\right)\log(1/\epsilon))$.

We saw that SDCA is like SGD since g_t is an unbiased stochastic gradient. Does SDCA really have lower variance than SGD? Yes! If you accept that the theorem statement is true (independent proof), then we can show that the variance provably goes to zero:

$$\begin{aligned}\mathbb{E}[\|g_t\|^2] &= \mathbb{E}\|\nabla f_i(w^{(t-1)}) + \alpha_i^* - \alpha_i^* + \alpha_i^{(t-1)}\|^2 \\ &\leq 2\mathbb{E}\|\nabla f_i(w^{(t-1)}) + \alpha_i^*\|^2 + 2\mathbb{E}\| - \alpha_i^* + \alpha_i^{(t-1)}\|^2\end{aligned}$$

Since $\alpha_i^* = -\nabla f_i(w^*)$, the first term $\|\nabla f_i(w^{(t-1)}) - \nabla f_i(w^*)\|^2 \leq L^2\|w^{(t-1)} - w^*\|^2$. The theorem then implies that both the first and the second terms go to zero (and hence the variance of the stochastic gradient term approaches zero - more like SGD at the start and more like GD at the end!).

SVRG. SVRG is slightly more complex to introduce, because there is an inner and outer loop. But here's the main idea. When $F(w) = \frac{1}{n}\sum_{i=1}^n F_i(w)$,

$$\begin{aligned}\nabla F(w) &= \nabla F(w) - \nabla F(\tilde{w}) + \nabla F(\tilde{w}) \\ &\approx \nabla F_i(w) - \nabla F_i(\tilde{w}) + \nabla F(\tilde{w}) := g_t\end{aligned}$$

where $\tilde{w}$ is some nearby point to w (we batch-calculate $\nabla F(\tilde{w})$ once in $O(n)$ steps and then use it for the next $O(n)$ stochastic steps). In other words, instead of approximating $\nabla F(w)$ the gradient by $\nabla F_i(w)$, we approximate it by something which is still unbiased but with lower variance. Indeed, using a similar argument as before, the authors show that $\mathbb{E}\|g_t\|^2 \leq 4L(F(w) - F(w^*) + F(\tilde{w}) - F(w^*)) \to 0$.

5 Optimality of SDCA/SVRG/SAGA/...?

Since each iteration takes time d, the total complexity of SDCA is $\left(\frac{L}{\lambda} + n\right) d\log(1/\epsilon)$. Compare this to vanilla non-stochastic gradient descent, which needs nd time per iteration, finishing in $\frac{L}{\lambda}\log(1/\epsilon)$ iterations, giving a total complexity of $n\frac{L}{\lambda}d\log(1/\epsilon)$. Hence SDCA is much better than both a naive stochastic algorithm like SGD and a purely deterministic algorithm like GD. Note: accelerated GD has better dependence on condition number κ.

Note that the above convergence rate is great, but it is not provably optimal, since one can derive an *accelerated A-SDCA* algorithm, which can then be further extended to an accelerated and proximal AP-SDCA. Even that is not optimal, until we have matching lower bounds. Agarwal and Bottou recently proved a (non-matching) lower bound for minimizing finite sums of $O(n + \sqrt{n(\kappa - 1)}\log(1/\epsilon))$ calls to an incremental first order oracle (which makes sense, because as $n \to \infty$, you get back to usual SGD model with high variance and shouldn't get a linear rate, and for the second term if $f_i = f$, then there is no variance and by Nesterov's lower bounds it should get worse by at least $\sqrt{\kappa - 1}\log(1/\epsilon)$), and this leaves three natural possibilities:

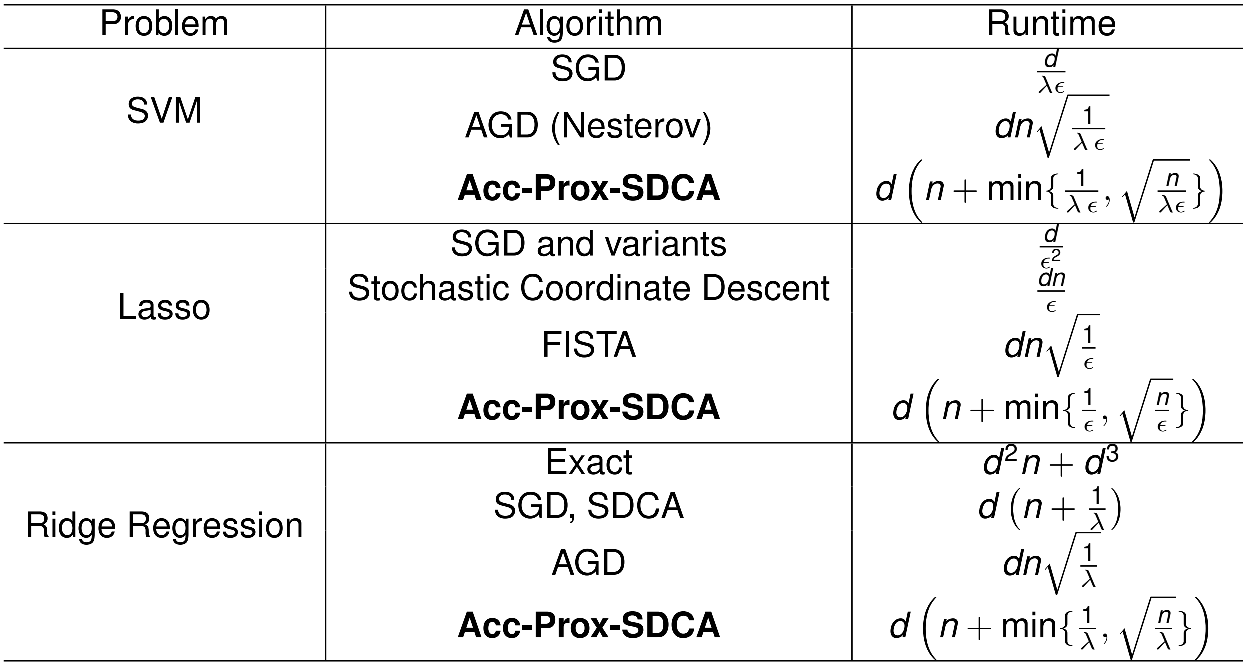
1. The algorithms may not be optimal, or not optimal in all ranges of κ.

2. Maybe these algorithms are indeed optimal - one may be able to prove even tighter upper bounds for A-SDCA (or variants of SVRG or SAGA or ...).
3. One may be able to prove tighter lower bounds.

Agarwal and Bottou also make a very important argument — it is not easy to compare batch and incremental/stochastic methods immediately. This is because the strong convexity constant of F can be much larger than the strong convexity constant of f_i, which was caused by the regularization parameter λ (like for overcomplete ridge regression), and the Lipschitz constant of the gradient of F can be smaller than the Lipschitz constant for f_i, denoted by L. Hence, even though methods like AP-SDCA are optimal in the worst case sense (for arbitrary f_i), current analysis can still sometimes place them worse than Nesterov's Accelerated Gradient Descent.

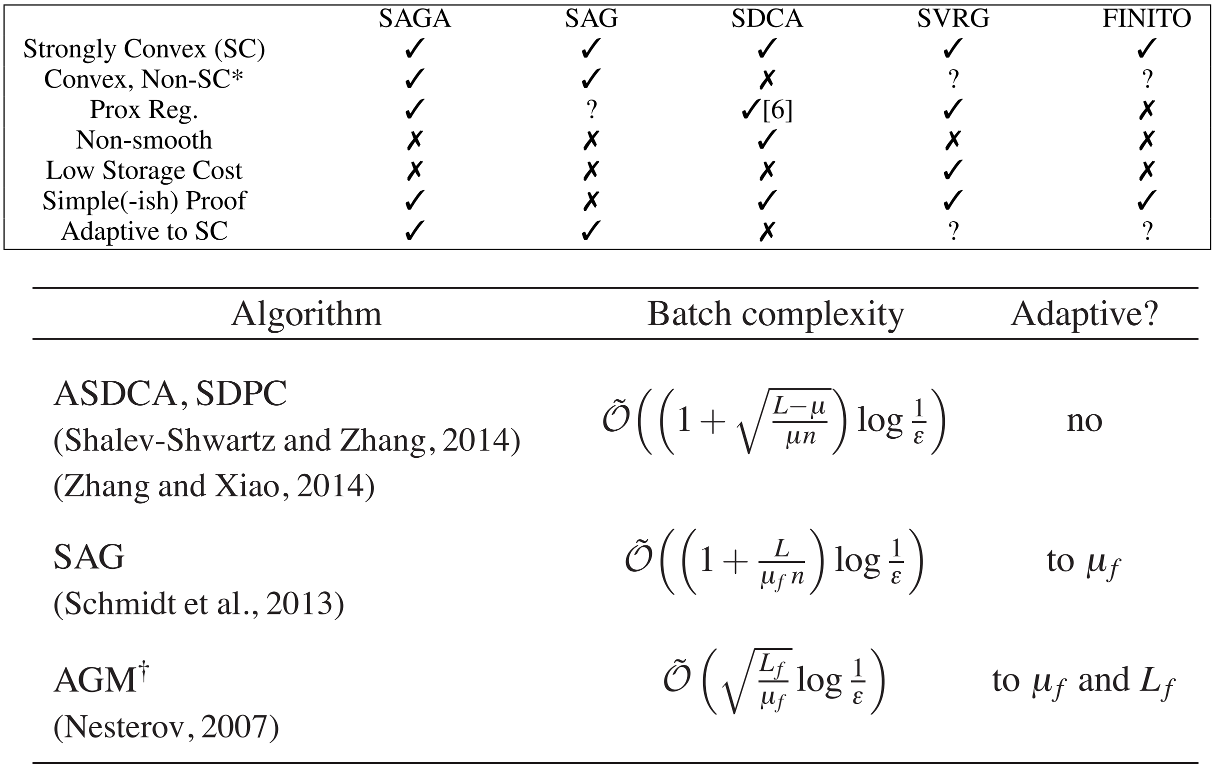
Time will definitely settle the issue, this research area is very active. Extensive experiments will be needed, as well as better analysis where each f_i is related to other f_is by a distributional assumption on the underlying data.

6 Comparing Methods



Problem	Algorithm	Runtime
SVM	SGD	$\frac{d}{\lambda\epsilon}$
	AGD (Nesterov)	$dn\sqrt{\frac{1}{\lambda\,\epsilon}}$
	Acc-Prox-SDCA	$d\left(n+\min\{\frac{1}{\lambda\,\epsilon},\sqrt{\frac{n}{\lambda\epsilon}}\}\right)$
Lasso	SGD and variants	$\frac{d}{\epsilon^2}$
	Stochastic Coordinate Descent	$\frac{dn}{\epsilon}$
	FISTA	$dn\sqrt{\frac{1}{\epsilon}}$
	Acc-Prox-SDCA	$d\left(n+\min\{\frac{1}{\epsilon},\sqrt{\frac{n}{\epsilon}}\}\right)$
Ridge Regression	Exact	d^2n+d^3
	SGD, SDCA	$d\left(n+\frac{1}{\lambda}\right)$
	AGD	$dn\sqrt{\frac{1}{\lambda}}$
	Acc-Prox-SDCA	$d\left(n+\min\{\frac{1}{\lambda},\sqrt{\frac{n}{\lambda}}\}\right)$

Figure 1: Top figure is from Tong Zhang's slides on accelerated proximal SDCA, middle from the SAGA paper, bottom from Agarwal and Bottou's lower bounds paper.



	SAGA	SAG	SDCA	SVRG	FINITO
Strongly Convex (SC)	✓	✓	✓	✓	✓
Convex, Non-SC*	✓	✓	✗	?	?
Prox Reg.	✓	?	✓[6]	✓	✗
Non-smooth	✗	✗	✓	✗	✗
Low Storage Cost	✗	✗	✗	✓	✗
Simple(-ish) Proof	✓	✗	✓	✓	✓
Adaptive to SC	✓	✓	✗	?	?

Algorithm	Batch complexity	Adaptive?
ASDCA, SDPC (Shalev-Shwartz and Zhang, 2014) (Zhang and Xiao, 2014)	$\tilde{\mathcal{O}}\left(\left(1+\sqrt{\frac{L-\mu}{\mu n}}\right)\log\frac{1}{\varepsilon}\right)$	no
SAG (Schmidt et al., 2013)	$\tilde{\mathcal{O}}\left(\left(1+\frac{L}{\mu_f n}\right)\log\frac{1}{\varepsilon}\right)$	to μ_f
AGM† (Nesterov, 2007)	$\tilde{\mathcal{O}}\left(\sqrt{\frac{L_f}{\mu_f}}\log\frac{1}{\varepsilon}\right)$	to μ_f and L_f

Figure 2: Top figure is from Tong Zhang's slides on accelerated proximal SDCA, middle from the SAGA paper, bottom from Agarwal and Bottou's lower bounds paper.