Finito: A Faster, Permutable Incremental Gradient Method for Big Data Problems

Aaron J. Defazio AARON.DEFAZIO@ANU.EDU.AU
Tibério S. Caetano TIBERIO.CAETANO@NICTA.COM.AU
Justin Domke JUSTIN.DOMKE@NICTA.COM.AU
NICTA and Australian National University

Abstract

Recent advances in optimization theory have shown that smooth strongly convex finite sums can be minimized faster than by treating them as a black box ”batch” problem. In this work we introduce a new method in this class with a theoretical convergence rate four times faster than existing methods, for sums with sufficiently many terms. This method is also amendable to a sampling without replacement scheme that in practice gives further speed-ups. We give empirical results showing state of the art performance.

1. Introduction

Many recent advances in the theory and practice of numerical optimization have come from the recognition and exploitation of structure. Perhaps the most common structure is that of finite sums. In machine learning when applying empirical risk minimization we almost always end up with an optimization problem involving the minimization of a sum with one term per data point.

The recently developed SAG algorithm (Schmidt et al., 2013) has shown that even with this simple form of structure, as long as we have sufficiently many data points we are able to do significantly better than black-box optimization techniques in expectation for smooth strongly convex problems. In practical terms the difference is often a factor of 10 or more.

The requirement of sufficiently large datasets is fundamental to these methods. We describe the precise form of this as the big data condition. Essentially, it is the requirement that the amount of data is on the same order as the condition number of the problem. The strong convexity requirement is not as onerous. Strong convexity holds in the common case where a quadratic regularizer is used together with a convex loss.

The SAG method and the Finito method we describe in this work are similar in their form to stochastic gradient descent methods, but with one crucial difference: They store additional information about each data point during optimization. Essentially, when they revisit a data point, they do not treat it as a novel piece of information every time.

Methods for the minimization of finite sums have classically been known as Incremental gradient methods (Bertsekas, 2010). The proof techniques used in SAG differ fundamentally from those used on other incremental gradient methods though. The difference hinges on the requirement that data be accessed in a randomized order. SAG does not work when data is accessed sequentially each epoch, so any proof technique which shows even non-divergence for sequential access cannot be applied.

A remarkable property of Finito is the tightness of the theoretical bounds compared to the practical performance of the algorithm. The practical convergence rate seen is at most twice as good as the theoretically predicted rate. This sets it apart from methods such as LBFGS where the empirical performance is often much better than the relatively weak theoretical convergence rates would suggest.

The lack of tuning required also sets Finito apart from stochastic gradient descent (SGD). In order to get good performance out of SGD, substantial laborious tuning of multiple constants has traditionally been required. A multitude of heuristics have been developed to help choose these constants, or adapt them as the method progresses. Such heuristics are more complex than Finito, and do not have the same theoretical backing. SGD has application outside of convex problems of course, and we do not propose that Finito will replace SGD in those settings. Even on strongly convex problems SGD does not exhibit linear convergence like Finito does.

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

There are many similarities between SAG, Finto and stochastic dual coordinate descent (SDCA) methods (Shalev-Shwartz & Zhang, 2013). SDCA is only applicable to linear predictors. When it can be applied, it has linear convergence with theoretical rates similar to SAG and Finito.

2. Algorithm

We consider differentiable convex functions of the form

$$f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w).$$

We assume that each f_i has Lipschitz continuous gradients with constant L and is strongly convex with constant s. Clearly if we allow $n = 1$, virtually all smooth, strongly convex problems are included. So instead, we will restrict ourselves to problems satisfying the *big data* condition.

Big data condition: Functions of the above form satisfy the big data condition with constant β if

$$n \geq \beta \frac{L}{s}$$

Typical values of β are 1-8. In plain language, we are considering problems where the amount of data is of the same order as the condition number (L/s) of the problem.

2.1. Additional Notation

We superscript with (k) to denote the value of the scripted quantity at iteration k. We omit the n superscript on summations, and subscript with i with the implication that indexing starts at 1. When we use separate arguments for each f_i, we denote them ϕ_i. Let $\bar{\phi}^{(k)}$ denote the average $\bar{\phi}^{(k)} = \frac{1}{n}\sum_i^n \phi_i^{(k)}$. Our step length constant, which depends on β, is denoted α. We use angle bracket notation for dot products $\langle \cdot, \cdot \rangle$.

2.2. The Finito algorithm

We start with a table of known $\phi_i^{(0)}$ values, and a table of known gradients $f_i'(\phi_i^{(0)})$, for each i. We will update these two tables during the course of the algorithm. The step for iteration k, is as follows:

1. Update w using the step:
$$w^{(k)} = \bar{\phi}^{(k)} - \frac{1}{\alpha s n}\sum_i f_i'(\phi_i^{(k)}).$$
2. Pick an index j uniformly at random, or using without-replacement sampling as discussed in Section 3.
3. Set $\phi_j^{(k+1)} = w^{(k)}$ in the table and leave the other variables the same ($\phi_i^{(k+1)} = \phi_i^{(k)}$ for $i \neq j$).
4. Calculate and store $f_j'(\phi_j^{(k+1)})$ in the table.

Our main theoretical result is a convergence rate proof for this method.

Theorem 1. *When the big data condition holds with $\beta = 2$, $\alpha = 2$ may be used. In that setting, if we have initialized all $\phi_i^{(0)}$ the same, the convergence rate is:*

$$E\left[f(\bar{\phi}^{(k)})\right] - f(w^*) \leq \frac{3}{4s}\left(1 - \frac{1}{2n}\right)^k \left\| f'(\bar{\phi}^{(0)}) \right\|^2.$$

See Section 5 for the proof. In contrast, SAG achieves a $\left(1 - \frac{1}{8n}\right)$ rate when $\beta = 2$. Note that on a per epoch basis, the Finito rate is $\left(1 - \frac{1}{2n}\right)^n \approx \exp(-1/2) = 0.606$. To put that into context, 10 epochs will see the error bound reduced by more than 148x.

One notable feature of our method is the fixed step size. In typical machine learning problems the strong convexity constant is given by the strength constant of the quadratic regularizer used. Since this is a known quantity, as long as the big data condition holds $\alpha = 2$ may be used without any tuning or adjustment of Finito required. This lack of tuning is a major feature of Finito.

In cases where the big data condition does not hold, we conjecture that the step size must be reduced proportionally to the violation of the big data condition. In practice, the most effective step size can be found by testing a number of step sizes, as is usually done with other stochastic optimisation methods.

A simple way of satisfying the big data condition is to duplicate your data enough times so then holds. This is not as effective in practice as just changing the step size, and of course it uses more memory. However it does fall within the current theory.

Another difference compared to the SAG method is that we store both gradients and points ϕ_i. We do not actually need twice as much memory however as they can be stored summed together. In particular we store the quantities $p_i = f_i'(\phi_i) - \alpha s \phi_i$, and use the update rule $w = -\frac{1}{\alpha s n}\sum_i p_i$. This trick does not work when step lengths are adjusted during optimization however. The storage of ϕ_i is also a disadvantage when the gradients $f_i'(\phi_i)$ are sparse but ϕ_i are not sparse, as it can cause significant additional memory usage. We do not recommend the usage of Finito when gradients are sparse.

The SAG algorithm differs from Finito only in the w update and step lengths:

$$w^{(k)} = w^{(k-1)} - \frac{1}{16Ln}\sum_i f_i'(\phi_i^{(k)}).$$

3. Randomness is key

By far the most interesting aspect of the SAG and Finito methods is the random choice of index at each iteration. We are not in an online setting, so there is no inherent randomness in the problem. Yet it seems that a randomized method is required. Neither method works in practice when the same ordering is used each pass, or in fact with any non-random access scheme we have tried. It is hard to emphasize enough the importance of randomness here. The technique of pre-permuting the data, then doing in order passes after that, also does not work. Reducing the step size in SAG or Finito by 1 or 2 orders of magnitude does not fix the convergence issues either.

Other methods, such as standard SGD, have been noted by various authors to exhibit speed-ups when random sampling is used instead of in order passes, but the differences are not as extreme as convergence v.s. non-convergence. Perhaps the most similar problem is that of coordinate descent on smooth convex functions. Coordinate descent cannot diverge when non-random orderings are used, but convergence rates are substantially worse in the non-randomized setting (Nesterov 2010, Richtarik & Takac 2011).

Reducing the step size α by a much larger amount, namely by a factor of n, does allow for non-randomized orderings to be used. This gives an *extremely* slow method however. This is the case covered by the MISO (Mairal, 2013). A similar reduction in step size gives convergence under non-randomized orderings for SAG also. Convergence rates for incremental sub-gradient methods with a variety of orderings appear in the literature also (Nedic & Bertsekas, 2000).

Sampling without replacement is much faster

Other sampling schemes, such as sampling without replacement, should be considered. In detail, we mean the case where each ”pass” over the data is a set of sampling without replacement steps, which continue until no data remains, after which another ”pass” starts afresh. We call this the permuted case for simplicity, as it is the same as re-permuting the data after each pass. In practice, this approach does not give any speedup with SAG, however it works spectacularly well with Finito. We see speedups of up to a factor of two using this approach. This is one of the major differences in practice between SAG and Finito. We should note that we have no theory to support this case however. We are not aware of any analysis that proves faster convergence rates of any optimization method under a sampling without replacement scheme. An interesting discussion of SGD under without-replacement sampling appears in Recht & Re (2012).

The SDCA method is also sometimes used with a permuted ordering (Shalev-Shwartz & Zhang, 2013), our experiments in Section 7 show that this sometimes results in a large speedup over uniform random sampling, although it does not appear to be as reliable as with Finito.

4. Proximal variant

We now consider composite problems of the form

$$f(w) = \frac{1}{n}\sum_i f_i(w) + \lambda r(w),$$

where r is convex but not necessarily smooth or strongly convex. Such problems are often addressed using proximal algorithms, particularly when the proximal operator for r:

$$\text{prox}_\lambda^r(z) = \text{argmin}_x \quad \frac{1}{2}\|x-z\|^2 + \lambda r(x)$$

has a closed form solution. An example would be the use of L1 regularization. We now describe the Finito update for this setting. First notice that when we set w in the Finito method, it can be interpreted as minimizing the quantity:

$$B(x) = \frac{1}{n}\sum_i f_i(\phi_i) + \frac{1}{n}\sum_i \langle f_i'(\phi_i), x-\phi_i\rangle + \frac{\alpha s}{2n}\sum_i \|x-\phi_i\|^2,$$

with respect to x, for fixed ϕ_i. This is related to the upper bound minimized by MISO, where αs is instead L. It is straight forward to modify this for the composite case:

$$B_{\lambda r}(x) = \lambda r(x) + \frac{1}{n}\sum_i f_i(\phi_i) + \frac{1}{n}\sum_i \langle f_i'(\phi_i), x-\phi_i\rangle + \frac{\alpha s}{2n}\sum_i \|x-\phi_i\|^2.$$

The minimizer of the modified $B_{\lambda r}$ can be expressed using the proximal operator as:

$$w = \text{prox}_{\lambda/\alpha s}^r\left(\bar{\phi} - \frac{1}{\alpha s n}\sum_i f_i'(\phi_i)\right).$$

This strongly resembles the update in the standard gradient descent setting, which for a step size of $1/L$ is

$$w = \text{prox}_{\lambda/L}^r\left(w^{(k-1)} - \frac{1}{L}f'(w^{(k-1)})\right).$$

We have not yet developed any theory supporting the proximal variant of Finito, although empirical evidence suggests it has the same convergence rate as in the non-proximal case.

5. Convergence proof

We start by stating two simple lemmas. All expectations in the following are over the choice of index j at step k. Quantities without superscripts are at their values at iteration k.

Lemma 1. *The expected step is*

$$E[w^{(k+1)}] - w = -\frac{1}{\alpha s n} f'(w).$$

I.e. the w step is a gradient descent step in expectation ($\frac{1}{\alpha s n} \propto \frac{1}{L}$). A similar equality also holds for SGD, but not for SAG.

Proof.

$$\begin{aligned}
&E[w^{(k+1)}] - w \\
&\quad = E\left[\frac{1}{n}(w - \phi_j) - \frac{1}{\alpha s n}\left(f'_j(w) - f'_j(\phi_j)\right)\right] \\
&\quad = \frac{1}{n}(w - \bar{\phi}) - \frac{1}{\alpha s n} f'(w) + \frac{1}{\alpha s n^2}\sum_i f'_i(\phi_i)
\end{aligned}$$

Now simplify $\frac{1}{n}(w - \bar{\phi})$ as $-\frac{1}{\alpha s n^2}\sum_i f'_i(\phi_i)$, so the only term that remains is $-\frac{1}{\alpha s n} f'(w)$. □

Lemma 2. *(Decomposition of variance) We can decompose $\frac{1}{n}\sum_i \|w - \phi_i\|^2$ as*

$$\frac{1}{n}\sum_i \|w - \phi_i\|^2 = \|w - \bar{\phi}\|^2 + \frac{1}{n}\sum_i \|\bar{\phi} - \phi_i\|^2.$$

Proof.

$$\begin{aligned}
&\frac{1}{n}\sum_i \|w - \phi_i\|^2 \\
&= \|w - \bar{\phi}\|^2 + \frac{1}{n}\sum_i \|\bar{\phi} - \phi_i\|^2 + \frac{2}{n}\sum_i \left\langle w - \bar{\phi}, \bar{\phi} - \phi_i\right\rangle \\
&= \|w - \bar{\phi}\|^2 + \frac{1}{n}\sum_i \|\bar{\phi} - \phi_i\|^2 + 2\left\langle w - \bar{\phi}, \bar{\phi} - \bar{\phi}\right\rangle \\
&= \|w - \bar{\phi}\|^2 + \frac{1}{n}\sum_i \|\bar{\phi} - \phi_i\|^2.
\end{aligned}$$

□

Main proof

Our proof proceeds by construction of a Lyapunov function T; that is, a function that bounds a quantity of interest, and that decreases each iteration in expectation. Our Lyapunov function $T = T_1 + T_2 + T_3 + T_4$ is composed of the sum of the following terms,

$$T_1 = f(\bar{\phi}),$$

$$T_2 = -\frac{1}{n}\sum_i f_i(\phi_i) - \frac{1}{n}\sum_i \left\langle f'_i(\phi_i), w - \phi_i\right\rangle,$$

$$T_3 = -\frac{s}{2n}\sum_i \|w - \phi_i\|^2,$$

$$T_4 = \frac{s}{2n}\sum_i \|\bar{\phi} - \phi_i\|^2.$$

We now state how each term changes between steps $k+1$ and k. Proofs are found in the appendix in the supplementary material:

$$E[T_1^{(k+1)}] - T_1 \le \frac{1}{n}\left\langle f'(\bar{\phi}), w - \bar{\phi}\right\rangle + \frac{L}{2n^3}\sum_i \|w - \phi_i\|^2,$$

$$\begin{aligned}
E[T_2^{(k+1)}] - T_2 \le{}& -\frac{1}{n}T_2 - \frac{1}{n}f(w) \\
&+ \left(\frac{1}{\alpha} - \frac{\beta}{n}\right)\frac{1}{sn^3}\sum_i \|f'_i(w) - f'_i(\phi_i)\|^2 \\
&+ \frac{1}{n}\left\langle \bar{\phi} - w, f'(w)\right\rangle \\
&- \frac{1}{n^3}\sum_i \left\langle f'_i(w) - f'_i(\phi_i), w - \phi_i\right\rangle,
\end{aligned}$$

$$\begin{aligned}
E[T_3^{(k+1)}] - T_3 ={}& -\left(\frac{1}{n} + \frac{1}{n^2}\right)T_3 + \frac{1}{\alpha n}\left\langle f'(w), w - \bar{\phi}\right\rangle \\
&- \frac{1}{2\alpha^2 s n^3}\sum_i \|f'_i(\phi_i) - f'_i(w)\|^2,
\end{aligned}$$

$$\begin{aligned}
E[T_4^{(k+1)}] - T_4 ={}& -\frac{s}{2n^2}\sum_i \|\bar{\phi} - \phi_i\|^2 + \frac{s}{2n}\|\bar{\phi} - w\|^2 \\
&- \frac{s}{2n^3}\sum_i \|w - \phi_i\|^2.
\end{aligned}$$

Theorem 2. *Between steps k and $k+1$, if $\frac{2}{\alpha} - \frac{1}{\alpha^2} - \beta + \frac{\beta}{\alpha} \le 0$, $\alpha \ge 2$ and $\beta \ge 2$ then*

$$E[T^{(k+1)}] - T \le -\frac{1}{\alpha n}T.$$

Proof. We take the three lemmas above and group like terms to get

$$\begin{aligned}
E[T^{(k+1)}] - T \le{}& \frac{1}{n}\left\langle f'(\bar{\phi}), w - \bar{\phi}\right\rangle + \frac{1}{n^2}\sum_i f_i(\phi_i) \\
&- \frac{1}{n}f(w) + \frac{1}{n^2}\sum_i \left\langle f'_i(\phi_i), w - \phi_i\right\rangle \\
&+ \left(1 - \frac{1}{\alpha}\right)\frac{1}{n}\left\langle f'(w), \bar{\phi} - w\right\rangle \\
&+ \left(\frac{L}{sn} + 1\right)\frac{s}{2n^2}\sum_i \|w - \phi_i\|^2 \\
&- \frac{1}{n^3}\sum_i \left\langle f'_i(w) - f'_i(\phi_i), w - \phi_i\right\rangle \\
&+ \left(1 - \frac{1}{2\alpha}\right)\frac{1}{\alpha s n^3}\sum_i \|f'_i(\phi_i) - f'_i(w)\|^2
\end{aligned}$$

$$+\frac{s}{2n}\left\|w-\bar{\phi}\right\|^2-\frac{s}{2n^2}\sum_i\left\|\bar{\phi}-\phi_i\right\|^2.$$

Next we cancel part of the first line using

$$\frac{1}{\alpha n}\left\langle f'(\bar{\phi}),w-\bar{\phi}\right\rangle\leq\frac{1}{\alpha n}f(w)-\frac{1}{\alpha n}f(\bar{\phi})-\frac{s}{2\alpha n}\left\|w-\bar{\phi}\right\|^2,$$

based on B3 in the Appendix. We then pull terms occurring in $-\frac{1}{\alpha n}T$ together, giving $E[T^{(k+1)}]-T\leq$

$$-\frac{1}{\alpha n}T+(1-\frac{1}{\alpha})\frac{1}{n}\left\langle f'(\bar{\phi})-f'(w),w-\bar{\phi}\right\rangle$$
$$+(1-\frac{1}{\alpha})\left[-\frac{1}{n}f(w)-\frac{1}{n}T_2\right]$$
$$+(\frac{L}{sn}+1-\frac{1}{\alpha})\frac{s}{2n^2}\sum_i\left\|w-\phi_i\right\|^2$$
$$-\frac{1}{n^3}\sum_i\left\langle f_i'(w)-f_i'(\phi_i),w-\phi_i\right\rangle$$
$$+(1-\frac{1}{2\alpha})\frac{1}{\alpha sn^3}\sum_i\left\|f_i'(\phi_i)-f_i'(w)\right\|^2$$
$$+(1-\frac{1}{\alpha})\frac{s}{2n}\left\|w-\bar{\phi}\right\|^2-(1-\frac{1}{\alpha})\frac{s}{2n^2}\sum_i\left\|\bar{\phi}-\phi_i\right\|^2.$$

Next we use the standard inequality (B5)

$$(1-\frac{1}{\alpha})\frac{1}{n}\left\langle f'(\bar{\phi})-f'(w),w-\bar{\phi}\right\rangle\leq-(1-\frac{1}{\alpha})\frac{s}{n}\left\|w-\bar{\phi}\right\|^2,$$

which changes the bottom row to $-(1-\frac{1}{\alpha})\frac{s}{2n}\left\|w-\bar{\phi}\right\|^2-(1-\frac{1}{\alpha})\frac{s}{2n^2}\sum_i\left\|\bar{\phi}-\phi_i\right\|^2$. These two terms can then be grouped using Lemma 2, to give

$$E[T^{(k+1)}]-T\leq-\frac{1}{\alpha n}T+\frac{L}{2n^3}\sum_i\left\|w-\phi_i\right\|^2$$
$$+(1-\frac{1}{\alpha})\left[-\frac{1}{n}f(w)-\frac{1}{n}T_2\right]$$
$$-\frac{1}{n^3}\sum_i\left\langle f_i'(w)-f_i'(\phi_i),w-\phi_i\right\rangle$$
$$+(1-\frac{1}{2\alpha})\frac{1}{\alpha sn^3}\sum_i\left\|f_i'(\phi_i)-f_i'(w)\right\|^2.$$

We use the following inequality (Corollary 6 in Appendix) to cancel against the $\sum_i\left\|w-\phi_i\right\|^2$ term:

$$\frac{1}{\beta}\left[-\frac{1}{n}f(w)-\frac{1}{n}T_2\right]\leq\frac{1}{n^3}\sum_i\left\langle f_i'(w)-f_i'(\phi_i),w-\phi_i\right\rangle$$
$$-\frac{L}{2n^3}\sum_i\left\|w-\phi_i\right\|^2$$
$$-\frac{1}{2sn^3}\sum_i\left\|f_i'(w)-f_i'(\phi_i)\right\|^2,$$

and then apply the following similar inequality (B7 in Appendix) to partially cancel $\sum_i\left\|f_i(\phi_i)-f_i(w)\right\|^2$:

$$\left(1-\frac{1}{\alpha}-\frac{1}{\beta}\right)\left[-\frac{1}{n}f(w)-\frac{1}{n}T_2\right]$$
$$\leq-\left(1-\frac{1}{\alpha}-\frac{1}{\beta}\right)\frac{\beta}{2sn^3}\sum_i\left\|f_i'(\phi_i)-f_i'(w)\right\|^2.$$

Leaving us with

$$E[T^{(k+1)}]-T\leq-\frac{1}{\alpha n}T$$
$$+(\frac{2}{\alpha}-\frac{1}{\alpha^2}-\beta+\frac{\beta}{\alpha})\frac{1}{2sn^3}\sum_i\left\|f_i'(\phi_i)-f_i'(w)\right\|^2.$$

The remaining gradient norm term is non-positive under the conditions specified in our assumptions. □

Theorem 3. *The Lyapunov function bounds* $f(\bar{\phi})-f(w^*)$ *as follows:*

$$f(\bar{\phi}^{(k)})-f(w^*)\leq\alpha T^{(k)}.$$

Proof. Consider the following function, which we will call $R(x)$:

$$R(x)=\frac{1}{n}\sum_i f_i(\phi_i)+\frac{1}{n}\sum_i\left\langle f_i'(\phi_i),x-\phi_i\right\rangle$$
$$+\frac{s}{2n}\sum_i\left\|x-\phi_i\right\|^2.$$

When evaluated at its minimum with respect to x, which we denote $w'=\bar{\phi}-\frac{1}{sn}\sum_i f_i'(\phi_i)$, it is a lower bound on $f(w^*)$ by strong convexity. However, we are evaluating at $w=\bar{\phi}-\frac{1}{\alpha sn}\sum_i f_i'(\phi_i)$ instead in the (negated) Lyapunv function. R is convex with respect to x, so by definition

$$R(w)=R\left(\left(1-\frac{1}{\alpha}\right)\bar{\phi}+\frac{1}{\alpha}w'\right)$$
$$\leq\left(1-\frac{1}{\alpha}\right)R(\bar{\phi})+\frac{1}{\alpha}R(w').$$

Therefore by the lower bounding property

$$f(\bar{\phi})-R(w)\geq f(\bar{\phi})-\left(1-\frac{1}{\alpha}\right)R(\bar{\phi})-\frac{1}{\alpha}R(w')$$
$$\geq f(\bar{\phi})-\left(1-\frac{1}{\alpha}\right)f(\bar{\phi})-\frac{1}{\alpha}f(w^*)$$
$$=\frac{1}{\alpha}\left(f(\bar{\phi})-f(w^*)\right).$$

Now note that $T\geq f(\bar{\phi})-R(w)$. So

$$f(\bar{\phi})-f(w^*)\leq\alpha T.$$

□

Theorem 4. *If the Finito method is initialized with all* $\phi_i^{(0)}$ *the same,and the assumptions of Theorem 2 hold, then the* *convergence rate is:*

$$E\left[f(\bar{\phi}^{(k)})\right] - f(w^*) \leq \frac{c}{s}\left(1 - \frac{1}{\alpha n}\right)^k \left\|f'(\bar{\phi}^{(0)})\right\|^2,$$

with $c = \left(1 - \frac{1}{2\alpha}\right)$.

Proof. By unrolling Theorem 2, we get

$$E[T^{(k)}] \leq \left(1 - \frac{1}{\alpha n}\right)^k T^{(0)}.$$

Now using Theorem 3

$$E\left[f(\bar{\phi}^{(k)})\right] - f(w^*) \leq \alpha\left(1 - \frac{1}{\alpha n}\right)^k T^{(0)}.$$

We need to control $T^{(0)}$ also. Since we are assuming that all ϕ_i^0 start the same, we have that

$$\begin{aligned}T^{(0)} &= f(\bar{\phi}^{(0)}) - \frac{1}{n}\sum_i f_i(\bar{\phi}^{(0)}) \\ &\quad - \frac{1}{n}\sum_i \left\langle f_i'(\bar{\phi}^{(0)}), w^{(0)} - \bar{\phi}^{(0)}\right\rangle - \frac{s}{2}\left\|w^{(0)} - \bar{\phi}^{(0)}\right\|^2 \\ &= 0 - \left\langle f'(\bar{\phi}^{(0)}), w^{(0)} - \bar{\phi}^{(0)}\right\rangle - \frac{s}{2}\left\|-\frac{1}{\alpha s}f'(\bar{\phi}^{(0)})\right\|^2 \\ &= \frac{1}{\alpha s}\left\|f'(\bar{\phi}^{(0)})\right\|^2 - \frac{1}{2\alpha^2 s}\left\|f'(\bar{\phi}^{(0)})\right\|^2 \\ &= \left(1 - \frac{1}{2\alpha}\right)\frac{1}{\alpha s}\left\|f'(\bar{\phi}^{(0)})\right\|^2.\end{aligned}$$

□

6. Lower complexity bounds and exploiting problem structure

The theory for the class of smooth, strongly convex problems with Lipschitz continuous gradients under first order optimization methods (known as $S_{s,L}^{1,1}$) is well developed. These results require the technical condition that the dimensionality of the input space R^m is much larger than the number of iterations we will take. For simplicity we will assume this is the case in the following discussions.

It is known that problems exist in $S_{s,L}^{1,1}$ for which the iterate convergence rate is bounded by:

$$\left\|w^{(k)} - w^*\right\|^2 \geq \left(\frac{\sqrt{L/s} - 1}{\sqrt{L/s} + 1}\right)^{2k}\left\|w^{(0)} - w^*\right\|^2.$$

In fact, when s and L are known in advance, this rate is achieved up to a small constant factor by several methods, most notably by Nesterov's accelerated gradient descent method (Nesterov 1988, Nesterov 1998). In order to achieve convergence rates faster than this, additional assumptions must be made on the class of functions considered.

Recent advances have shown that all that is required to achieve significantly faster rates is a finite sum structure, such as in our problem setup. When the big data condition holds our method achieves a rate 0.6065 per epoch in expectation. This rate only depends on the condition number indirectly, through the big data condition. For example, with $L/s = 1,000,000$, the fastest possible rate for a black box method is a 0.996, whereas Finito achieves a rate of 0.6065 in expectation for $n \geq 4,000,000$, or 124x faster. The required amount of data is not unusual in modern machine learning problems. In practice, when quasi-newton methods are used instead of accelerated methods, a speedup of 10-20x is more common.

6.1. Oracle class

We now describe the (stochastic) oracle class $FS_{s,L,n}^{1,1}(R^m)$ for which SAG and Finito most naturally fit.

Function class: $f(w) = \frac{1}{n}\sum_{i=1}^n f_i(w)$, with $f_i \in S_{s,L}^{1,1}(R^m)$.

Oracle: Each query takes a point $x \in R^m$, and returns j, $f_j(w)$ and $f_j'(w)$, with j chosen uniformly at random.

Accuracy: Find w such that $E[\left\|w^{(k)} - w^*\right\|^2] \leq \epsilon$.

The main choice made in formulating this definition is putting the random choice in the oracle. This restricts the methods allowed quite strongly. The alternative case, where the index j is input to the oracle in addition to x, is also interesting. Assuming that the method has access to a source of true random indices, we call that class $DS_{s,L,n}^{1,1}(R^m)$. In Section 3 we discuss empirical evidence that suggests that faster rates are possible in $DS_{s,L,n}^{1,1}(R^m)$ than for $FS_{s,L,n}^{1,1}(R^m)$.

It should first be noted that there is a trivial lower bound rate for $f \in SS_{s,L,\beta}^{1,1}(R^m)$ of $\left(1 - \frac{1}{n}\right)$ reduction per step. Its not clear if this can be achieved for any finite β. Finito is only a factor of 2 off this rate, namely $\left(1 - \frac{1}{2n}\right)$ at $\beta = 2$, and asymptotes towards this rate for very large β. SDCA, while not applicable to all problems in this class, also achieves the rate asymptotically.

Another case to consider is the smooth convex but non-strongly convex setting. We still assume Lipschitz continuous gradients. In this setting we will show that for sufficiently high dimensional input spaces, the (non-stochastic) lower complexity bound is the same for the finite sum case and cannot be better than that given by treating f as a single black box function.

The full proof is in the Appendix, but the idea is as follows: when the f_i are not strongly convex, we can choose them such that they do not interact with each other, as long as the dimensionality is much larger than k. More precisely, we may choose them so that for any x and y and any $i \neq j$, $\langle f_i'(x), f_j'(y)\rangle = 0$ holds. When the functions do not interact, no optimization scheme may reduce the iterate error faster than by just handling each f_i separately. Doing so in an in-order fashion gives the same rate as just treating f using a black box method.

For strongly convex f_i, it is not possible for them to not interact in the above sense. By definition strong convexity requires a quadratic component in each f_i that acts on all dimensions.

7. Experiments

In this section we compare Finito, SAG, SDCA and LBFGS. We only consider problems where the regularizer is large enough so that the big data condition holds, as this is the case our theory supports. However, in practice our method can be used with smaller step sizes in the more general case, in much the same way as SAG.

Since we do not know the Lipschitz constant for these problems exactly, the SAG method was run for a variety of step sizes, with the one that gave the fastest rate of convergence plotted. The best step-size for SAG is usually not what the theory suggests. Schmidt et al. (2013) suggest using $\frac{1}{L}$ instead of the theoretical rate $\frac{1}{16L}$. For Finito, we find that using $\alpha = 2$ is the fastest rate when the big data condition holds for any $\beta > 1$. This is the step suggested by our theory when $\beta = 2$. Interestingly, reducing α to 1 does not improve the convergence rate. Instead we see no further improvement in our experiments.

For both SAG and Finito we used a differing step rule than suggested by the theory for the first pass. For Finito, during the first pass, since we do not have derivatives for each ϕ_i yet, we simply sum over the k terms seen so far

$$w^{(k)} = \frac{1}{k}\sum_i^k \phi_i^{(k)} - \frac{1}{\alpha s k}\sum_i^k f_i'(\phi_i^{(k)}),$$

where we process data points in index order for the first pass only. A similar trick is suggested by Schmidt et al. (2013) for SAG.

Since SDCA only applies to linear predictors, we are restricted in possible test problems. We choose log loss for 3 binary classification datasets, and quadratic loss for 2 regression tasks. For classification, we tested on the ijcnn1 and covtype datasets[1], as well as MNIST[2] classifying 0-4 against 5-9. For regression, we choose the two datasets from the UCI repository: the million song year regression dataset, and the slice-localization dataset. The training portion of the datasets are of size 5.3×10^5, 5.0×10^4, 6.0×10^4, 4.7×10^5 and 5.3×10^4 respectively.

Figure 6 shows the results of our experiments. Firstly we can see that LBFGS is not competitive with any of the incremental gradient methods considered. Secondly, the non-permuted SAG, Finito and SDCA often converge at very similar rates. The observed differences are usually down to the speed of the very first pass, where SAG and Finito are using the above mentioned trick to speed their convergence. After the first pass, the slopes of the line are usually comparable. When considering the methods with permutation each pass, we see a clear advantage for Finito. Interestingly, it gives very flat lines, indicating very stable convergence.

8. Related work

Traditional incremental gradient methods (Bertsekas, 2010) have the same form as SGD, but applied to finite sums. Essentially they are the non-online analogue of SGD. Applying SGD to strongly convex problems does not yield linear convergence, and in practice it is slower than the linear-converging methods we discuss in the remainder of this section.

Besides the methods that fall under the classical Incremental gradient moniker, SAG and MISO (Mairal, 2013) methods are also related. MISO method falls into the class of upper bound minimization methods, such as EM and classical gradient descent. MISO is essentially the Finito method, but with step sizes n times smaller. When using these larger step sizes, the method is no longer a upper bound minimization method. Our method can be seen as MISO, but with a step size scheme that gives neither a lower nor upper bound minimisation method. While this work was under peer review, a tech report (**?**) was put on arXiv that establishes the convergence rate of MISO with step $\alpha = 1$ and with $\beta = 2$ as $1 - \frac{1}{3n}$ per step. This similar but not quite as good as the $1 - \frac{1}{2n}$ rate we establish.

Stochastic Dual Coordinate descent (Shalev-Shwartz & Zhang, 2013) also gives fast convergence rates on problems for which it is applicable. It requires computing the convex conjugate of each f_i, which makes it more complex to implement. For the best performance it has to take advantage of the structure of the losses also. For simple linear classification and regression problems it can be effective. When using a sparse dataset, it is a better choice than Finito due to the memory requirements. For linear predictors, its theoretical convergence rate of $\left(1 - \frac{\beta}{(1+\beta)n}\right)$ per step is a little faster than what we establish for Finito, however it does not appear to be faster in our experiments.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

[2] http://yann.lecun.com/exdb/mnist/

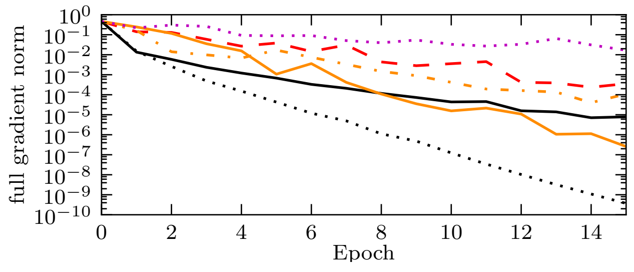

Figure 1. MNIST

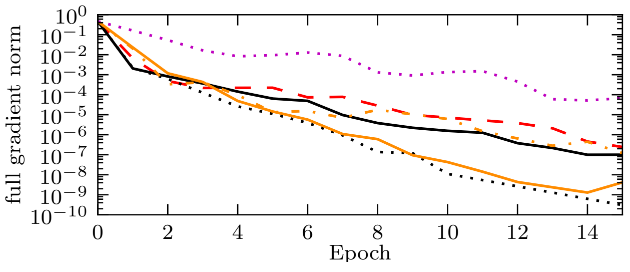

Figure 2. ijcnn1

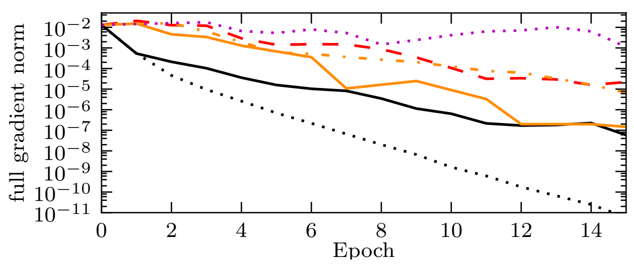

Figure 3. Covtype

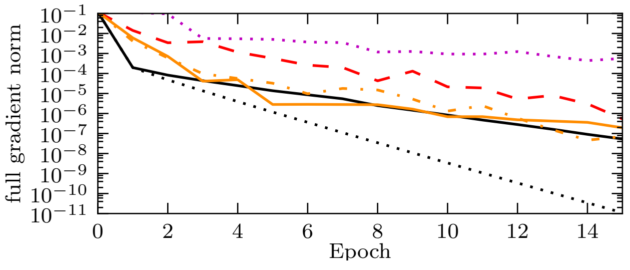

Figure 4. Million Song

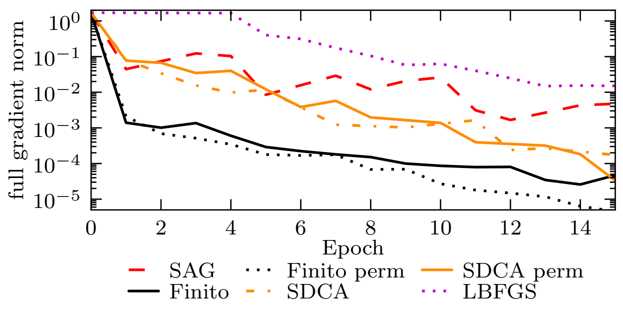

Figure 5. slice

Figure 6. Convergence rate plots for test problems

9. Conclusion

We have presented a new method for minimization of finite sums of smooth strongly convex functions, when there is a sufficiently large number of terms in the summation. We additionally develop some theory for the lower complexity bounds on this class, and show the empirical performance of our method.

References

Bertsekas, Dimitri P. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. Technical report, 2010.

Mairal, Julien. Optimization with first-order surrogate functions. *ICML*, 2013.

Nedic, Angelia and Bertsekas, Dimitri. *Stochastic Optimization: Algorithms and Applications*, chapter Convergence Rate of Incremental Subgradient Algorithms. Kluwer Academic, 2000.

Nesterov, Yu. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Mateaticheskie Metody*, 24:509–517, 1988.

Nesterov, Yu. *Introductory Lectures On Convex Programming*. Springer, 1998.

Nesterov, Yu. Efficiency of coordinate descent methods on huge-scale optimization problems. Technical report, CORE, 2010.

Recht, Benjamin and Re, Christopher. Beneath the valley of the noncommutative arithmetic-geometric mean inequality: conjectures, case-studies, and consequences. Technical report, University of Wisconsin-Madison, 2012.

Richtarik, Peter and Takac, Martin. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Technical report, University of Edinburgh, 2011.

Schmidt, Mark, Roux, Nicolas Le, and Bach, Francis. Minimizing finite sums with the stochastic average gradient. Technical report, INRIA, 2013.

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