
Breaking Locality Accelerates Block Gauss-Seidel

Stephen Tu¹ Shivaram Venkataraman¹ Ashia C. Wilson¹ Alex Gittens² Michael I. Jordan¹ Benjamin Recht¹

Abstract

Recent work by Nesterov and Stich (2016) showed that momentum can be used to accelerate the rate of convergence for block Gauss-Seidel in the setting where a fixed partitioning of the coordinates is chosen ahead of time. We show that this setting is too restrictive, constructing instances where breaking locality by running non-accelerated Gauss-Seidel with randomly sampled coordinates substantially outperforms accelerated Gauss-Seidel with any fixed partitioning. Motivated by this finding, we analyze the accelerated block Gauss-Seidel algorithm in the random coordinate sampling setting. Our analysis captures the benefit of acceleration with a new data-dependent parameter which is well behaved when the matrix sub-blocks are well-conditioned. Empirically, we show that accelerated Gauss-Seidel with random coordinate sampling provides speedups for large scale machine learning tasks when compared to non-accelerated Gauss-Seidel and the classical conjugate-gradient algorithm.

1. Introduction

The randomized Gauss-Seidel method is a commonly used iterative algorithm to compute the solution of an $n \times n$ linear system $Ax = b$ by updating a single coordinate at a time in a randomized order. While this approach is known to converge linearly to the true solution when A is positive definite (see e.g. (Leventhal & Lewis, 2010)), in practice it is often more efficient to update a small block of coordinates at a time due to the effects of cache locality.

In extending randomized Gauss-Seidel to the block setting, a natural question that arises is how one should sample the next block. At one extreme a *fixed partition* of the coordi-

nates is chosen ahead of time. The algorithm is restricted to randomly selecting blocks from this fixed partitioning, thus favoring data locality. At the other extreme we break locality by sampling a new set of *random coordinates* to form a block at every iteration.

Theoretically, the fixed partition case is well understood both for Gauss-Seidel (Qu et al., 2015; Gower & Richtárik, 2015) and its Nesterov accelerated variant (Nesterov & Stich, 2016). More specifically, at most $O(\mu_{\text{part}}^{-1} \log(1/\varepsilon))$ iterations of Gauss-Seidel are sufficient to reach a solution with at most ε error, where μ_{part} is a quantity which measures how well the A matrix is preconditioned by the block diagonal matrix containing the sub-blocks corresponding to the fixed partitioning. When acceleration is used, Nesterov and Stich (2016) show that the rate improves to $O\left(\sqrt{\frac{n}{p} \mu_{\text{part}}^{-1}} \log(1/\varepsilon)\right)$, where p is the partition size.

For the random coordinate selection model, the existing literature is less complete. While it is known (Qu et al., 2015; Gower & Richtárik, 2015) that the iteration complexity with random coordinate section is $O(\mu_{\text{rand}}^{-1} \log(1/\varepsilon))$ for an ε error solution, μ_{rand} is another instance dependent quantity which is not directly comparable to μ_{part} . Hence it is not obvious how much better, if at all, one expects random coordinate selection to perform compared to fixed partitioning.

Our first contribution in this paper is to show that, when compared to the random coordinate selection model, the fixed partition model can perform very poorly in terms of iteration complexity to reach a pre-specified error. Specifically, we present a family of instances (similar to the matrices recently studied by Lee and Wright (2016)) where *non-accelerated* Gauss-Seidel with random coordinate selection performs *arbitrarily* faster than both non-accelerated and even accelerated Gauss-Seidel, using *any* fixed partition. Our result thus shows the importance of the sampling strategy and that acceleration cannot make up for a poor choice of sampling distribution.

This finding motivates us to further study the benefits of acceleration under the random coordinate selection model. Interestingly, the benefits are more nuanced under this model. We show that acceleration improves the rate from $O(\mu_{\text{rand}}^{-1} \log(1/\varepsilon))$ to $O\left(\sqrt{\nu \mu_{\text{rand}}^{-1}} \log(1/\varepsilon)\right)$, where ν is

¹UC Berkeley, Berkeley, California, USA ²Rensselaer Polytechnic Institute, Troy, New York, USA. Correspondence to: Stephen Tu <stephent@berkeley.edu>.

a new instance dependent quantity that satisfies $\nu \leq \mu_{\text{rand}}^{-1}$. We derive a bound on ν which suggests that if the sub-blocks of A are all well conditioned, then acceleration can provide substantial speedups. We note that this is merely a sufficient condition, and our experiments suggest that our bound is conservative.

In the process of deriving our results, we also develop a general proof framework for randomized accelerated methods based on Wilson et al. (2016) which avoids the use of estimate sequences in favor of an explicit Lyapunov function. Using our proof framework we are able to recover recent results (Nesterov & Stich, 2016; Allen-Zhu et al., 2016) on accelerated coordinate descent. Furthermore, our proof framework allows us to immediately transfer our results on Gauss-Seidel over to the randomized accelerated Kaczmarz algorithm, extending a recent result by Liu and Wright (2016) on updating a single constraint at a time to the block case.

Finally, we empirically demonstrate that despite its theoretical nuances, accelerated Gauss-Seidel using random coordinate selection can provide significant speedups in practical applications over Gauss-Seidel with fixed partition sampling, as well as the classical conjugate-gradient (CG) algorithm. As an example, for a kernel ridge regression (KRR) task in machine learning on the augmented CIFAR-10 dataset ($n = 250,000$), acceleration with random coordinate sampling performs up to $1.5\times$ faster than acceleration with a fixed partitioning to reach an error tolerance of 10^{-2} , with the gap substantially widening for smaller error tolerances. Furthermore, it performs over $3.5\times$ faster than conjugate-gradient on the same task.

2. Background

We assume that we are given an $n \times n$ matrix A which is positive definite, and an n dimensional response vector b . We also fix an integer p which denotes a block size. Under the assumption of A being positive definite, the function $f(x) = \frac{1}{2}x^\top Ax - x^\top b$ is strongly convex and smooth. Recent analysis of Gauss-Seidel (Gower & Richtárik, 2015) proceeds by noting the connection between Gauss-Seidel and (block) coordinate descent on f . This is the point of view we will take in this paper.

2.1. Existing rates for randomized block Gauss-Seidel

We first describe the sketching framework of (Qu et al., 2015; Gower & Richtárik, 2015) and show how it yields rates on Gauss-Seidel when blocks are chosen via a fixed partition or randomly at every iteration. While we will only focus on the special case when the sketch matrix represents column sampling, the sketching framework allows us to provide a unified analysis of both cases.

To be more precise, let \mathcal{D} be a distribution over $\mathbb{R}^{n \times p}$, and let $S_k \sim \mathcal{D}$ be drawn iid from \mathcal{D} . If we perform block coordinate descent by minimizing f along the range of S_k , then the randomized block Gauss-Seidel update is given by

$$x_{k+1} = x_k - S_k(S_k^\top AS_k)^\dagger S_k^\top (Ax_k - b). \quad (1)$$

Column sampling. Every index set $J \subseteq [n]$ with $|J| = p$ induces a sketching matrix $S(J) = (e_{J(1)}, \dots, e_{J(p)})$ where e_i denotes the i -th standard basis vector in \mathbb{R}^n , and $J(1), \dots, J(p)$ is any ordering of the elements of J . By equipping different probability measures on $2^{[n]}$, one can easily describe fixed partition sampling as well as random coordinate sampling (and many other sampling schemes). The former puts uniform mass on the index sets $J_1, \dots, J_{n/p}$, whereas the latter puts uniform mass on all $\binom{n}{p}$ index sets of size p . Furthermore, in the sketching framework there is no limitation to use a uniform distribution, nor is there any limitation to use a fixed p for every iteration. For this paper, however, we will restrict our attention to these cases.

Existing rates. Under the assumptions stated above, (Qu et al., 2015; Gower & Richtárik, 2015) show that for every $k \geq 0$, the sequence (1) satisfies

$$\mathbb{E}[\|x_k - x_*\|_A] \leq (1 - \mu)^{k/2} \|x_0 - x_*\|_A, \quad (2)$$

where $\mu = \lambda_{\min}(\mathbb{E}[P_{A^{1/2}S}])$. The expectation in (2) is taken with respect to the randomness of S_0, S_1, \dots , and the expectation in the definition of μ is taken with respect to $S \sim \mathcal{D}$. Under both fixed partitioning and random coordinate selection, $\mu > 0$ is guaranteed (see e.g. (Gower & Richtárik, 2015), Lemma 4.3). Thus, (1) achieves a linear rate of convergence to the true solution, with the rate governed by the μ quantity shown above.

We now specialize (2) to fixed partitioning and random coordinate sampling, and provide some intuition for why we expect the latter to outperform the former in terms of iteration complexity. We first consider the case when the sampling distribution corresponds to fixed partitioning. Assume for notational convenience that the fixed partitioning corresponds to placing the first p coordinates in the first partition J_1 , the next p coordinates in the second partition J_2 , and so on. Here, $\mu = \mu_{\text{part}}$ corresponds to a measure of how close the product of A with the inverse of the block diagonal is to the identity matrix, defined as

$$\mu_{\text{part}} = \frac{p}{n} \lambda_{\min} \left(A \cdot \text{blkdiag} \left(A_{J_1}^{-1}, \dots, A_{J_{n/p}}^{-1} \right) \right). \quad (3)$$

Above, A_{J_i} denotes the $p \times p$ matrix corresponding to the sub-matrix of A indexed by the i -th partition. A loose lower bound on μ_{part} is

$$\mu_{\text{part}} \geq \frac{p}{n} \frac{\lambda_{\min}(A)}{\max_{1 \leq i \leq n/p} \lambda_{\max}(A_{J_i})}. \quad (4)$$

On the other hand, in the random coordinate case, Qu et al. (2015) derive a lower bound on $\mu = \mu_{\text{rand}}$ as

$$\mu_{\text{rand}} \geq \frac{p}{n} \left(\beta + (1 - \beta) \frac{\max_{1 \leq i \leq n} A_{ii}}{\lambda_{\min}(A)} \right)^{-1}, \quad (5)$$

where $\beta = (p - 1)/(n - 1)$. Using the lower bounds (4) and (5), we can upper bound the iteration complexity of fixed partition Gauss-Seidel N_{part} by $O\left(\frac{n}{p} \frac{\max_{1 \leq i \leq n/p} \lambda_{\max}(A_{J_i})}{\lambda_{\min}(A)} \log(1/\varepsilon)\right)$ and random coordinate Gauss-Seidel N_{rand} as $O\left(\frac{n}{p} \frac{\max_{1 \leq i \leq n} A_{ii}}{\lambda_{\min}(A)} \log(1/\varepsilon)\right)$. Comparing the bound on N_{part} to the bound on N_{rand} , it is not unreasonable to expect that random coordinate sampling has better iteration complexity than fixed partition sampling in certain cases. In Section 3, we verify this by constructing instances A such that fixed partition Gauss-Seidel takes arbitrarily more iterations to reach a pre-specified error tolerance compared with random coordinate Gauss-Seidel.

2.2. Accelerated rates for fixed partition Gauss-Seidel

Based on the interpretation of Gauss-Seidel as block coordinate descent on the function f , we can use Theorem 1 of Nesterov and Stich (2016) to recover a procedure and a rate for accelerating (1) in the fixed partition case; the specific details are discussed in the full version of the paper (Tu et al., 2017). We will refer to this procedure as ACDM.

The convergence guarantee of the ACDM procedure is that for all $k \geq 0$,

$$\mathbb{E}[\|x_k - x_*\|_A] \leq O\left(\left(1 - \sqrt{\frac{p}{n} \mu_{\text{part}}}\right)^{k/2} D_0\right). \quad (6)$$

Above, $D_0 = \|x_0 - x_*\|_A$ and μ_{part} is the same quantity defined in (3). Comparing (6) to the non-accelerated Gauss-Seidel rate given in (2), we see that acceleration improves the iteration complexity to reach a solution with ε error from $O(\mu_{\text{part}}^{-1} \log(1/\varepsilon))$ to $O\left(\sqrt{\frac{n}{p} \mu_{\text{part}}^{-1}} \log(1/\varepsilon)\right)$, as discussed in Section 1.

3. Results

We now present the main results of the paper. Proofs can be found in the full version (Tu et al., 2017) of this paper.

3.1. Fixed partition vs random coordinate sampling

Our first result is to construct instances where Gauss-Seidel with fixed partition sampling runs arbitrarily slower than random coordinate sampling, even if acceleration is used.

Consider the family of $n \times n$ positive definite matrices \mathcal{A} given by $\mathcal{A} = \{A_{\alpha,\beta} : \alpha > 0, \alpha + \beta > 0\}$ with $A_{\alpha,\beta}$

defined as $A_{\alpha,\beta} = \alpha I + \frac{\beta}{n} \mathbf{1}_n \mathbf{1}_n^\top$. The family \mathcal{A} exhibits a crucial property that $\Pi^\top A_{\alpha,\beta} \Pi = A_{\alpha,\beta}$ for every $n \times n$ permutation matrix Π . Lee and Wright (2016) recently exploited this invariance to illustrate the behavior of cyclic versus randomized permutations in coordinate descent.

We explore the behavior of Gauss-Seidel as the matrices $A_{\alpha,\beta}$ become ill-conditioned. To do this, we consider a particular parameterization which holds the minimum eigenvalue equal to one and sends the maximum eigenvalue to infinity via the sub-family $\{A_{1,\beta}\}_{\beta>0}$. Our first proposition characterizes the behavior of Gauss-Seidel with fixed partitions on this sub-family.

Proposition 3.1. *Fix $\beta > 0$ and positive integers n, p, k such that $n = pk$. Let $\{J_i\}_{i=1}^k$ be any partition of $\{1, \dots, n\}$ with $|J_i| = p$, and denote $S_i \in \mathbb{R}^{n \times p}$ as the column selector for partition J_i . Suppose $S \in \mathbb{R}^{n \times p}$ takes on value S_i with probability $1/k$. For every $A_{1,\beta} \in \mathcal{A}$ we have that*

$$\mu_{\text{part}} = \frac{p}{n + \beta p}. \quad (7)$$

Next, we perform a similar calculation under the random column sampling model.

Proposition 3.2. *Fix $\beta > 0$ and integers n, p such that $1 < p < n$. Suppose each column of $S \in \mathbb{R}^{n \times p}$ is chosen uniformly at random from $\{e_1, \dots, e_n\}$ without replacement. For every $A_{1,\beta} \in \mathcal{A}$ we have that*

$$\mu_{\text{rand}} = \frac{p}{n + \beta p} + \frac{(p-1)\beta p}{(n-1)(n + \beta p)}. \quad (8)$$

The differences between (7) and (8) are striking. Let us assume that β is much larger than n . Then, we have that $\mu_{\text{part}} \approx 1/\beta$ for (7), whereas $\mu_{\text{rand}} \approx \frac{p-1}{n-1}$ for (8). That is, μ_{part} can be made arbitrarily smaller than μ_{rand} as β grows.

Our next proposition states that the rate of Gauss-Seidel from (2) is tight order-wise in that for any instance there always exists a starting point which saturates the bound.

Proposition 3.3. *Let A be an $n \times n$ positive definite matrix, and let S be a random matrix such that $\mu = \lambda_{\min}(\mathbb{E}[P_{A^{1/2}S}]) > 0$. Let x_* denote the solution to $Ax = b$. There exists a starting point $x_0 \in \mathbb{R}^n$ such that the sequence (1) satisfies for all $k \geq 0$,*

$$\mathbb{E}[\|x_k - x_*\|_A] \geq (1 - \mu)^k \|x_0 - x_*\|_A. \quad (9)$$

From (2) we see that Gauss-Seidel using random coordinates computes a solution x_k satisfying $\mathbb{E}[\|x_k - x_*\|_{A_{1,\beta}}] \leq \varepsilon$ in at most $k = O\left(\frac{n}{p} \log(1/\varepsilon)\right)$ iterations. On the other hand, Proposition 3.3 states that for any fixed partition, there exists an input x_0 such that $k = \Omega(\beta \log(1/\varepsilon))$ iterations are required to reach the same ε error tolerance.

Furthermore, the situation does not improve even if use ACDM from (Nesterov & Stich, 2016). Proposition 3.6, which we describe later, implies that for any fixed partition there exists an input x_0 such that $k = \Omega\left(\sqrt{\frac{n}{p}}\beta \log(1/\varepsilon)\right)$ iterations are required for ACDM to reach ε error. Hence as $\beta \rightarrow \infty$, the gap between random coordinate and fixed partitioning can be made arbitrarily large. These findings are numerically verified in Section 5.1.

3.2. A Lyapunov analysis of accelerated Gauss-Seidel and Kaczmarz

Motivated by our findings, our goal is to understand the behavior of accelerated Gauss-Seidel under random coordinate sampling. In order to do this, we establish a general framework from which the behavior of accelerated Gauss-Seidel with random coordinate sampling follows immediately, along with rates for accelerated randomized Kaczmarz (Liu & Wright, 2016) and the accelerated coordinate descent methods of (Nesterov & Stich, 2016) and (Allen-Zhu et al., 2016).

For conciseness, we describe a simpler version of our framework which is still able to capture both the Gauss-Seidel and Kaczmarz results, deferring the general version to the full version of the paper. Our general result requires a bit more notation, but follows the same line of reasoning.

Let H be a random $n \times n$ positive semi-definite matrix. Put $G = \mathbb{E}[H]$, and suppose that G exists and is positive definite. Furthermore, let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be strongly convex and smooth, and let μ denote the strong convexity constant of f w.r.t. the $\|\cdot\|_{G^{-1}}$ norm.

Consider the following sequence $\{(x_k, y_k, z_k)\}_{k \geq 0}$ defined by the recurrence

$$x_{k+1} = \frac{1}{1+\tau}y_k + \frac{\tau}{1+\tau}z_k, \quad (10a)$$

$$y_{k+1} = x_{k+1} - H_k \nabla f(x_{k+1}), \quad (10b)$$

$$z_{k+1} = z_k + \tau(x_{k+1} - z_k) - \frac{\tau}{\mu} H_k \nabla f(x_{k+1}), \quad (10c)$$

where H_0, H_1, \dots are independent realizations of H and τ is a parameter to be chosen. Following (Wilson et al., 2016), we construct a candidate Lyapunov function V_k for the sequence (10) defined as

$$V_k = f(y_k) - f_* + \frac{\mu}{2} \|z_k - x_*\|_{G^{-1}}^2. \quad (11)$$

The following theorem demonstrates that V_k is indeed a Lyapunov function for (x_k, y_k, z_k) .

Theorem 3.4. *Let f, G, H be as defined above. Suppose further that f has 1-Lipschitz gradients w.r.t. the $\|\cdot\|_{G^{-1}}$ norm, and for every fixed $x \in \mathbb{R}^n$,*

$$f(\Phi(x; H)) \leq f(x) - \frac{1}{2} \|\nabla f(x)\|_H^2, \quad (12)$$

holds for a.e. H , where $\Phi(x; H) = x - H \nabla f(x)$. Set τ in (10) as $\tau = \sqrt{\mu/\nu}$, with

$$\nu = \lambda_{\max} \left(\mathbb{E} \left[(G^{-1/2} H G^{-1/2})^2 \right] \right).$$

Then for every $k \geq 0$, we have

$$\mathbb{E}[V_k] \leq (1 - \tau)^k V_0.$$

We now proceed to specialize Theorem 3.4 to both the Gauss-Seidel and Kaczmarz settings.

3.2.1. ACCELERATED GAUSS-SEIDEL

Let $S \in \mathbb{R}^{n \times p}$ denote a random sketching matrix. As suggested in Section 2, we set $f(x) = \frac{1}{2} x^T A x - x^T b$ and put $H = S(S^T A S)^{\dagger} S^T$. Note that $G = \mathbb{E}[S(S^T A S)^{\dagger} S^T]$ is positive definite iff $\lambda_{\min}(\mathbb{E}[P_{A^{1/2} S}]) > 0$, and is hence satisfied for both fixed partition and random coordinate sampling (c.f. Section 2). Next, the fact that f is 1-Lipschitz w.r.t. the $\|\cdot\|_{G^{-1}}$ norm and the condition (12) are standard calculations. All the hypotheses of Theorem 3.4 are thus satisfied, and the conclusion is Theorem 3.5, which characterizes the rate of convergence for accelerated Gauss-Seidel (Algorithm 1).

Algorithm 1 Accelerated randomized block Gauss-Seidel.

Require: $A \in \mathbb{R}^{n \times n}$, $A \succ 0$, $b \in \mathbb{R}^n$, sketching matrices $\{S_k\}_{k=0}^{T-1} \subseteq \mathbb{R}^{n \times p}$, $x_0 \in \mathbb{R}^n$, $\mu \in (0, 1)$, $\nu \geq 1$.

- 1: Set $\tau = \sqrt{\mu/\nu}$.
- 2: Set $y_0 = z_0 = x_0$.
- 3: **for** $k = 0, \dots, T - 1$ **do**
- 4: $x_{k+1} = \frac{1}{1+\tau}y_k + \frac{\tau}{1+\tau}z_k$.
- 5: $H_k = S_k(S_k^T A S_k)^{\dagger} S_k^T$.
- 6: $y_{k+1} = x_{k+1} - H_k(Ax_{k+1} - b)$.
- 7: $z_{k+1} = z_k + \tau(x_{k+1} - z_k) - \frac{\tau}{\mu} H_k(Ax_{k+1} - b)$.
- 8: **end for**
- 9: **Return** y_T .

Theorem 3.5. *Let A be an $n \times n$ positive definite matrix and $b \in \mathbb{R}^n$. Let $x_* \in \mathbb{R}^n$ denote the unique vector satisfying $Ax_* = b$. Suppose each S_k , $k = 0, 1, 2, \dots$ is an independent copy of a random matrix $S \in \mathbb{R}^{n \times p}$. Put $\mu = \lambda_{\min}(\mathbb{E}[P_{A^{1/2} S}])$, and suppose the distribution of S satisfies $\mu > 0$. Invoke Algorithm 1 with μ and ν , where*

$$\nu = \lambda_{\max} \left(\mathbb{E} \left[(G^{-1/2} H G^{-1/2})^2 \right] \right), \quad (13)$$

with $H = S(S^T A S)^{\dagger} S^T$ and $G = \mathbb{E}[H]$. Then with $\tau = \sqrt{\mu/\nu}$, for all $k \geq 0$,

$$\mathbb{E}[\|y_k - x_*\|_A] \leq \sqrt{2}(1 - \tau)^{k/2} \|x_0 - x_*\|_A. \quad (14)$$

Note that in the setting of Theorem 3.5, by the definition of ν and μ , it is always the case that $\nu \leq 1/\mu$. Therefore,

the iteration complexity of acceleration is at least as good as the iteration complexity without acceleration.

We conclude our discussion of Gauss-Seidel by describing the analogue of Proposition 3.3 for Algorithm 1, which shows that our analysis in Theorem 3.5 is tight order-wise. The following proposition applies to ACDM as well; we show in the full version of the paper how ACDM can be viewed as a special case of Algorithm 1.

Proposition 3.6. *Under the setting of Theorem 3.5, there exists starting positions $y_0, z_0 \in \mathbb{R}^n$ such that the iterates $\{(y_k, z_k)\}_{k \geq 0}$ produced by Algorithm 1 satisfy*

$$\mathbb{E}[\|y_k - x_*\|_A + \|z_k - x_*\|_A] \geq (1 - \tau)^k \|y_0 - x_*\|_A.$$

3.2.2. ACCELERATED KACZMARZ

The argument for Theorem 3.5 can be slightly modified to yield a result for randomized accelerated Kaczmarz in the sketching framework, for the case of a consistent overdetermined linear system.

Specifically, suppose we are given an $m \times n$ matrix A which has full column rank, and $b \in \mathcal{R}(A)$. Our goal is to recover the unique x_* satisfying $Ax_* = b$. To do this, we apply a similar line of reasoning as (Lee & Sidford, 2013). We set $f(x) = \frac{1}{2}\|x - x_*\|_2^2$ and $H = P_{A^\top S}$, where S again is our random sketching matrix. At first, it appears our choice of f is problematic since we do not have access to f and ∇f , but a quick calculation shows that $H\nabla f(x) = (S^\top A)^\dagger S^\top (Ax - b)$. Hence, with $r_k = Ax_k - b$, the sequence (10) simplifies to

$$x_{k+1} = \frac{1}{1 + \tau} y_k + \frac{\tau}{1 + \tau} z_k, \quad (15a)$$

$$y_{k+1} = x_{k+1} - (S_k^\top A)^\dagger S_k^\top r_{k+1}, \quad (15b)$$

$$z_{k+1} = z_k + \tau(x_{k+1} - z_k) - \frac{\tau}{\mu} (S_k^\top A)^\dagger S_k^\top r_{k+1}. \quad (15c)$$

The remainder of the argument proceeds nearly identically, and leads to the following theorem.

Theorem 3.7. *Let A be an $m \times n$ matrix with full column rank, and $b = Ax_*$. Suppose each S_k , $k = 0, 1, 2, \dots$ is an independent copy of a random sketching matrix $S \in \mathbb{R}^{m \times p}$. Put $H = P_{A^\top S}$ and $G = \mathbb{E}[H]$. The sequence (15) with $\mu = \lambda_{\min}(\mathbb{E}[P_{A^\top S}])$, $\nu = \lambda_{\max}(\mathbb{E}[(G^{-1/2} H G^{-1/2})^2])$, and $\tau = \sqrt{\mu/\nu}$ satisfies for all $k \geq 0$,*

$$\mathbb{E}[\|y_k - x_*\|_2] \leq \sqrt{2}(1 - \tau)^{k/2} \|x_0 - x_*\|_2. \quad (16)$$

Specialized to the setting of (Liu & Wright, 2016) where each row of A has unit norm and is sampled uniformly at every iteration, it can be shown (Section A.5.1) that

$\nu \leq m$ and $\mu = \frac{1}{m} \lambda_{\min}(A^\top A)$. Hence, the above theorem states that the iteration complexity to reach ε error is $O\left(\frac{m}{\sqrt{\lambda_{\min}(A^\top A)}} \log(1/\varepsilon)\right)$, which matches Theorem 5.1 of (Liu & Wright, 2016) order-wise. However, Theorem 3.7 applies in general for any sketching matrix.

3.3. Specializing accelerated Gauss-Seidel to random coordinate sampling

We now instantiate Theorem 3.5 to random coordinate sampling. The μ quantity which appears in Theorem 3.5 is identical to the quantity appearing in the rate (2) of non-accelerated Gauss-Seidel. That is, the iteration complexity to reach tolerance ε is $O\left(\sqrt{\nu \mu_{\text{rand}}^{-1}} \log(1/\varepsilon)\right)$, and the only new term here is ν . In order to provide a more intuitive interpretation of the ν quantity, we present an upper bound on ν in terms of an effective block condition number defined as follows. Given an index set $J \subseteq [n]$, define the effective block condition number of a matrix A as $\kappa_{\text{eff}, J}(A) = \frac{\max_{i \in J} A_{ii}}{\lambda_{\min}(A_J)}$. Note that $\kappa_{\text{eff}, J}(A) \leq \kappa(A_J)$ always. The following lemma gives upper and lower bounds on the ν quantity.

Lemma 3.8. *Let A be an $n \times n$ positive definite matrix and let p satisfy $1 < p < n$. We have that*

$$\frac{n}{p} \leq \nu \leq \frac{n}{p} \left(\frac{p-1}{n-1} + \left(1 - \frac{p-1}{n-1}\right) \kappa_{\text{eff}, p}(A) \right),$$

where $\kappa_{\text{eff}, p}(A) = \max_{J \subseteq [n]: |J|=p} \kappa_{\text{eff}, J}(A)$, ν is defined in (13), and the distribution of S corresponds to uniformly selecting p coordinates without replacement.

Lemma 3.8 states that if the $p \times p$ sub-blocks of A are well-conditioned as defined by the effective block condition number $\kappa_{\text{eff}, J}(A)$, then the speed-up of accelerated Gauss-Seidel with random coordinate selection over its non-accelerate counterpart parallels the case of fixed partitioning sampling (i.e. the rate described in (6) versus the rate in (2)). This is a reasonable condition, since very ill-conditioned sub-blocks will lead to numerical instabilities in solving the sub-problems when implementing Gauss-Seidel. On the other hand, we note that Lemma 3.8 provides merely a sufficient condition for speed-ups from acceleration, and is conservative. Our numerical experiments in Section A.7.2 suggest that in many cases the ν parameter behaves closer to the lower bound n/p than Lemma 3.8 suggests.

We can now combine Theorem 3.5 with (5) to derive the following upper bound on the iteration complexity of accelerated Gauss-Seidel with random coordinates as

$$N_{\text{rand, acc}} \leq O\left(\frac{n}{p} \sqrt{\frac{\max_{1 \leq i \leq n} A_{ii}}{\lambda_{\min}(A)}} \kappa_{\text{eff}, p}(A) \log(1/\varepsilon)\right).$$

Illustrative example. We conclude our results by illustrating our bounds on a simple example. Consider the subfamily $\{A_\delta\}_{\delta>0} \subseteq \mathcal{A}$, with

$$A_\delta = A_{n+\delta, -n}, \quad \delta > 0. \quad (17)$$

A simple calculation yields that $\kappa_{\text{eff},p}(A_\delta) = \frac{n-1+\delta}{n-p+\delta}$, and hence Lemma 3.8 states that $\nu(A_\delta) \leq \frac{n}{p} \left(1 + \frac{p-1}{n-1}\right)$. Furthermore, by a similar calculation to Proposition 3.2, $\mu_{\text{rand}} = \frac{p\delta}{n(n-p+\delta)}$. Assuming for simplicity that $p = o(n)$ and $\delta \in (0, 1)$, Theorem 3.5 states that at most $O\left(\frac{n^{3/2}}{p\sqrt{\delta}} \log(1/\varepsilon)\right)$ iterations are sufficient for an ε -accurate solution. On the other hand, without acceleration (2) states that $O\left(\frac{n^2}{p\delta} \log(1/\varepsilon)\right)$ iterations are sufficient and Proposition 3.3 shows there exists a starting position for which it is necessary. Hence, as either n grows large or δ tends to zero, the benefits of acceleration become more pronounced.

4. Related Work

We split the related work into two broad categories of interest: (a) work related to coordinate descent (CD) methods on convex functions and (b) randomized solvers designed for solving consistent linear systems.

When A is positive definite, Gauss-Seidel can be interpreted as an instance of coordinate descent on a strongly convex quadratic function. We therefore review related work on both non-accelerated and accelerated coordinate descent, focusing on the randomized setting instead of the more classical cyclic order or Gauss-Southwell rule for selecting the next coordinate. See (Tseng & Yun, 2009) for a discussion on non-random selection rules, (Nutini et al., 2015) for a comparison of random selection versus Gauss-Southwell, and (Nutini et al., 2016) for efficient implementations of Gauss-Southwell.

Nesterov’s original paper in (2012) first considered randomized CD on convex functions, assuming a partitioning of coordinates fixed ahead of time. The analysis included both non-accelerated and accelerated variants for convex functions. This work sparked a resurgence of interest in CD methods for large problems. Most relevant to our paper are extensions to the block setting (Richtárik & Takáč, 2014), handling arbitrary sampling distributions (Qu & Richtárik, 2014a;b; Fountoulakis & Tappenden, 2016), and second order updates for quadratic functions (Qu et al., 2016).

For accelerated CD, Lee and Sidford (2013) generalize the analysis of Nesterov (2012). While the analysis of (Lee & Sidford, 2013) was limited to selecting a single coordinate at a time, several follow on works (Qu & Richtárik, 2014a; Lin et al., 2014; Lu & Xiao, 2015; Fercoq & Richtárik, 2015) generalize to block and non-smooth settings. More recently, both Allen-Zhu et al. (2016) and Nesterov and

Stich (2016) independently improve the results of (Lee & Sidford, 2013) by using a different non-uniform sampling distribution. One of the most notable aspects of the analysis in (Allen-Zhu et al., 2016) is a departure from the (probabilistic) estimate sequence framework of Nesterov. Instead, the authors construct a valid Lyapunov function for coordinate descent, although they do not explicitly mention this. In our work, we make this Lyapunov point of view explicit. The constants in our acceleration updates arise from a particular discretization and Lyapunov function outlined from Wilson et al. (2016). Using this framework makes our proof particularly transparent, and allows us to recover results for strongly convex functions from (Allen-Zhu et al., 2016) and (Nesterov & Stich, 2016) as a special case.

From the numerical analysis side both the Gauss-Seidel and Kaczmarz algorithm are classical methods. Strohmer and Vershynin (2009) were the first to prove a linear rate of convergence for randomized Kaczmarz, and Leventhal and Lewis (2010) provide a similar kind of analysis for randomized Gauss-Seidel. Both of these were in the single constraint/coordinate setting. The block setting was later analyzed by Needell and Tropp (2014). More recently, Gower and Richtárik (2015) provide a unified analysis for both randomized block Gauss-Seidel and Kaczmarz in the sketching framework. We adopt this framework in this paper. Finally, Liu and Wright (2016) provide an accelerated analysis of randomized Kaczmarz once again in the single constraint setting and we extend this to the block setting.

5. Experiments

In this section we experimentally validate our theoretical results on how our accelerated algorithms can improve convergence rates. Our experiments use a combination of synthetic matrices and matrices from large scale machine learning tasks.

Setup. We run all our experiments on a 4 socket Intel Xeon CPU E7-8870 machine with 18 cores per socket and 1TB of DRAM. We implement all our algorithms in Python using `numpy`, and use the Intel MKL library with 72 OpenMP threads for numerical operations. We report errors as relative errors, i.e. $\|x_k - x_*\|_A^2 / \|x_*\|_A^2$. Finally, we use the best values of μ and ν found by tuning each experiment.

We implement fixed partitioning by creating random blocks of coordinates at the beginning of the experiment and cache the corresponding matrix blocks to improve performance. For random coordinate sampling, we select a new block of coordinates at each iteration.

For our fixed partition experiments, we restrict our attention to uniform sampling. While Gower and Richtárik (2015) propose a non-uniform scheme based on $\text{Tr}(S^T A S)$, for translation-invariant kernels this reduces to

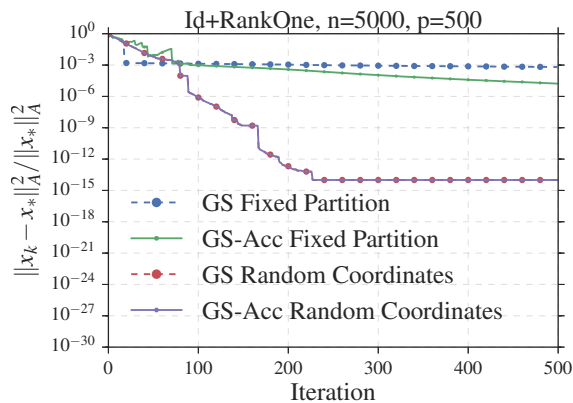


Figure 1. Experiments comparing fixed partitions versus random coordinate sampling for the example from Section 3.1 with $n = 5000$ coordinates, block size $p = 500$.

uniform sampling. Furthermore, as the kernel block Lipschitz constants were also roughly the same, other non-uniform schemes (Allen-Zhu et al., 2016) also reduce to nearly uniform sampling.

5.1. Fixed partitioning vs random coordinate sampling

Our first set of experiments numerically verify the separation between fixed partitioning sampling versus random coordinate sampling.

Figure 1 shows the progress per iteration on solving $A_{1,\beta}x = b$, with the $A_{1,\beta}$ defined in Section 3.1. Here we set $n = 5000$, $p = 500$, $\beta = 1000$, and $b \sim N(0, I)$. Figure 1 verifies our analytical findings in Section 3.1, that the fixed partition scheme is substantially worse than uniform sampling on this instance. It also shows that in this case, acceleration provides little benefit in the case of random coordinate sampling. This is because both μ and $1/\nu$ are order-wise p/n , and hence the rate for accelerated and non-accelerated coordinate descent coincide. However we note that this only applies for matrices where μ is as large as it can be (i.e. p/n), that is instances for which Gauss-Seidel is already converging at the optimal rate (see (Gower & Richtárik, 2015), Lemma 4.2).

5.2. Kernel ridge regression

We next evaluate how fixed partitioning and random coordinate sampling affects the performance of Gauss-Seidel on large scale machine learning tasks. We use the popular image classification dataset CIFAR-10 and evaluate a kernel ridge regression (KRR) task with a Gaussian kernel. Specifically, given a labeled dataset $\{(x_i, y_i)\}_{i=1}^n$, we solve the linear system $(K + \lambda I)\alpha = Y$ with $K_{ij} = \exp(-\gamma \|x_i - x_j\|_2^2)$, where $\lambda, \gamma > 0$ are tunable parameters. The key property of KRR is that the kernel matrix K is positive semi-definite, and hence Algorithm 1 applies.

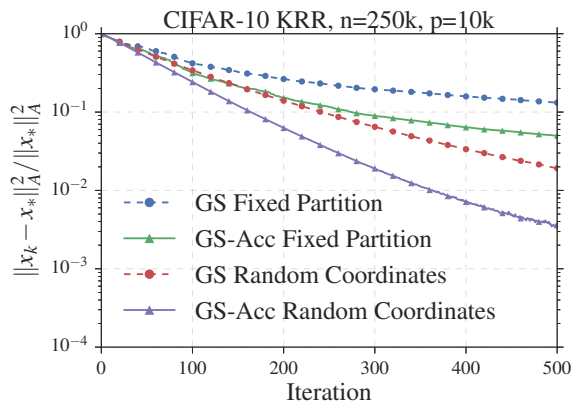


Figure 2. Experiments comparing fixed partitions versus uniform random sampling for CIFAR-10 augmented matrix while running kernel ridge regression. The matrix has $n = 250000$ coordinates and we set block size to $p = 10000$.

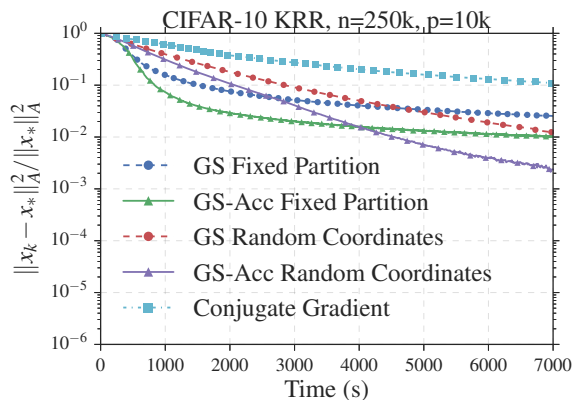


Figure 3. Comparing conjugate gradient with accelerated and un-accelerated Gauss-Seidel methods for CIFAR-10 augmented matrix while running kernel ridge regression. The matrix has $n = 250000$ coordinates and we set block size to $p = 10000$.

For the CIFAR-10 dataset, we augment the dataset¹ to include five reflections, translations per-image and then apply standard pre-processing steps used in image classification (Coates & Ng, 2012; Sparks et al., 2017). We finally apply a Gaussian kernel on our pre-processed images and the resulting kernel matrix has $n = 250000$ coordinates. We also include experiments on a smaller MNIST kernel matrix ($n = 60000$) in Section A.7.

Results from running 500 iterations of random coordinate sampling and fixed partitioning algorithms are shown in Figure 2. Comparing convergence across iterations, similar to previous section, we see that un-accelerated Gauss-Seidel with random coordinate sampling is better than accelerated Gauss-Seidel with fixed partitioning. However we also see that using acceleration with random sampling can further improve the convergence rates, especially to achieve errors of 10^{-3} or lower.

¹Similar to <https://github.com/akrizhevsky/cuda-convnet2>.

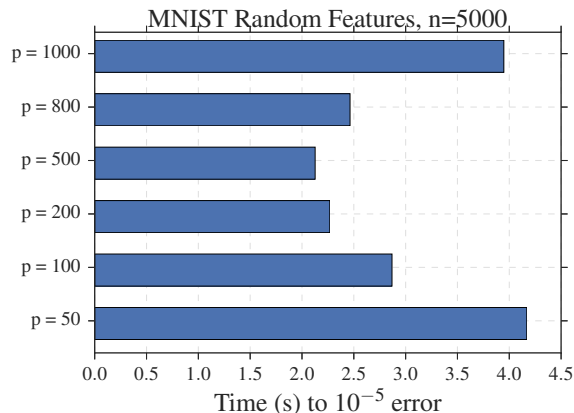


Figure 4. The effect of block size on the accelerated Gauss-Seidel method. For the MNIST dataset (pre-processed using random features) we see that block size of $p = 500$ works best.

We also compare the convergence with respect to running time in Figure 3. Fixed partitioning has better performance in practice random access is expensive in multi-core systems. However, we see that this speedup in implementation comes at a substantial cost in terms of convergence rate. For example in the case of CIFAR-10, using fixed partitions leads to an error of 1.2×10^{-2} after around 7000 seconds. In comparison we see that random coordinate sampling achieves a similar error in around 4500 seconds and is thus $1.5 \times$ faster. We also note that this speedup increases for lower error tolerances.

5.3. Comparing Gauss-Seidel to Conjugate-Gradient

We also compared Gauss-Seidel with random coordinate sampling to the classical conjugate-gradient (CG) algorithm. CG is an important baseline to compare with, as it is the de-facto standard iterative algorithm for solving linear systems in the numerical analysis community. While we report the results of CG without preconditioning, we remark that the performance using a standard banded preconditioner was not any better. However, for KRR specifically, there have been recent efforts (Avron et al., 2017; Rudi et al., 2017) to develop better preconditioners, and we leave a more thorough comparison for future work. The results of our experiment are shown in Figure 3. We note that Gauss-Seidel both with and without acceleration outperform CG. As an example, we note that to reach error 10^{-1} on CIFAR-10, CG takes roughly 7000 seconds, compared to less than 2000 seconds for accelerated Gauss-Seidel, which is a $3.5 \times$ improvement.

To understand this performance difference, we recall that our matrices A are fully dense, and hence each iteration of CG takes $O(n^2)$. On the other hand, each iteration of both non-accelerated and accelerated Gauss-Seidel takes $O(np^2 + p^3)$. Hence, as long as $p = O(n^{2/3})$, the time per iteration of Gauss-Seidel is order-wise no worse than

CG. In terms of iteration complexity, standard results state that CG takes at most $O(\sqrt{\kappa} \log(1/\varepsilon))$ iterations to reach an ε error solution, where κ denotes the condition number of A . On the other hand, Gauss-Seidel takes at most $O(\frac{n}{p} \kappa_{\text{eff}} \log(1/\varepsilon))$, where $\kappa_{\text{eff}} = \frac{\max_{1 \leq i \leq n} A_{ii}}{\lambda_{\min}(A)}$. In the case of any (normalized) kernel matrix associated with a translation-invariant kernel such as the Gaussian kernel, we have $\max_{1 \leq i \leq n} A_{ii} = 1$, and hence generally speaking κ_{eff} is much lower than κ .

5.4. Effect of block size

We next analyze the importance of the block size p for the accelerated Gauss-Seidel method. As the values of μ and ν change for each setting of p , we use a smaller MNIST matrix for this experiment. We apply a random feature transformation (Rahimi & Recht, 2007) to generate an $n \times d$ matrix F with $d = 5000$ features. We then use $A = F^T F$ and $b = F^T Y$ as inputs to the algorithm. Figure 4 shows the wall clock time to converge to 10^{-5} error as we vary the block size from $p = 50$ to $p = 1000$.

Increasing the block-size improves the amount of progress that is made per iteration but the time taken per iteration increases as $O(p^3)$ (Line 5, Algorithm 1). However, using efficient BLAS-3 primitives usually affords a speedup from systems techniques like cache blocking. We see the effects of this in Figure 4 where using $p = 500$ performs better than using $p = 50$. We also see that these benefits reduce for much larger block sizes and thus $p = 1000$ is slower.

6. Conclusion

In this paper, we extended the accelerated block Gauss-Seidel algorithm beyond fixed partition sampling. Our analysis introduced a new data-dependent parameter ν which governs the speed-up of acceleration. Specializing our theory to random coordinate sampling, we derived an upper bound on ν which shows that well conditioned blocks are a sufficient condition to ensure speedup. Experimentally, we showed that random coordinate sampling is readily accelerated beyond what our bound suggests.

The most obvious question remains to derive a sharper bound on the ν constant from Theorem 3.5. Another interesting question is whether or not the iteration complexity of random coordinate sampling is always bounded above by the iteration complexity with fixed coordinate sampling.

We also plan to study an implementation of accelerated Gauss-Seidel in a distributed setting (Tu et al., 2016). The main challenges here are in determining how to sample coordinates without significant communication overheads, and to efficiently estimate μ and ν . To do this, we wish to explore other sampling schemes such as shuffling the coordinates at the end of every epoch (Recht & Ré, 2013).

Acknowledgements

We thank Ross Boczar for assisting us with Mathematica support for non-commutative algebras, Orianna DeMasi for providing useful feedback on earlier drafts of this manuscript, and the anonymous reviewers for their helpful feedback. ACW is supported by an NSF Graduate Research Fellowship. BR is generously supported by ONR awards N00014-11-1-0723 and N00014-13-1-0129, NSF award CCF-1359814, the DARPA Fundamental Limits of Learning (Fun LoL) Program, a Sloan Research Fellowship, and a Google Research Award. This research is supported in part by DHS Award HSHQDC-16-3-00083, NSF CISE Expeditions Award CCF-1139158, DOE Award SN10040 DE-SC0012463, and DARPA XData Award FA8750-12-2-0331, and gifts from Amazon Web Services, Google, IBM, SAP, The Thomas and Stacey Siebel Foundation, Apple Inc., Arimo, Blue Goji, Bosch, Cisco, Cray, Cloudera, Ericsson, Facebook, Fujitsu, HP, Huawei, Intel, Microsoft, Mitre, Pivotal, Samsung, Schlumberger, Splunk, State Farm and VMware.

References

- Allen-Zhu, Zeyuan, Richtárik, Peter, Qu, Zheng, and Yuan, Yang. Even Faster Accelerated Coordinate Descent Using Non-Uniform Sampling. In *ICML*, 2016.
- Avron, Haim, Clarkson, Kenneth L., and Woodruff, David P. Faster Kernel Ridge Regression Using Sketching and Preconditioning. *arXiv*, 1611.03220, 2017.
- Coates, Adam and Ng, Andrew Y. Learning Feature Representations with K-Means. In *Neural Networks: Tricks of the Trade*. Springer, 2012.
- Fercoq, Olivier and Richtárik, Peter. Accelerated, Parallel, and Proximal Coordinate Descent. *SIAM J. Optim.*, 25(4), 2015.
- Fountoulakis, Kimon and Tappenden, Rachael. A Flexible Coordinate Descent Method. *arXiv*, 1507.03713, 2016.
- Gower, Robert M. and Richtárik, Peter. Randomized Iterative Methods for Linear Systems. *SIAM Journal on Matrix Analysis and Applications*, 36, 2015.
- Lee, Ching-Pei and Wright, Stephen J. Random Permutations Fix a Worst Case for Cyclic Coordinate Descent. *arXiv*, 1607.08320, 2016.
- Lee, Yin Tat and Sidford, Aaron. Efficient Accelerated Coordinate Descent Methods and Faster Algorithms for Solving Linear Systems. In *FOCS*, 2013.
- Leventhal, Dennis and Lewis, Adrian S. Randomized Methods for Linear Constraints: Convergence Rates and Conditioning. *Mathematics of Operations Research*, 35(3), 2010.
- Lin, Qihang, Lu, Zhaosong, and Xiao, Lin. An Accelerated Proximal Coordinate Gradient Method. In *NIPS*, 2014.
- Liu, Ji and Wright, Stephen J. An Accelerated Randomized Kaczmarz Algorithm. *Mathematics of Computation*, 85(297), 2016.
- Lu, Zhaosong and Xiao, Lin. On the Complexity Analysis of Randomized Block-Coordinate Descent Methods. *Mathematical Programming*, 152(1–2), 2015.
- Needell, Deanna and Tropp, Joel A. Paved with Good Intentions: Analysis of a Randomized Block Kaczmarz Method. *Linear Algebra and its Applications*, 441, 2014.
- Nesterov, Yurii. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM J. Optim.*, 22(2), 2012.
- Nesterov, Yurii and Stich, Sebastian. Efficiency of Accelerated Coordinate Descent Method on Structured Optimization Problems. Technical report, Université catholique de Louvain, CORE Discussion Papers, 2016.
- Nutini, Julie, Schmidt, Mark, Laradji, Issam H., Friedlander, Michael, and Koepke, Hoyt. Coordinate Descent Converges Faster with the Gauss-Southwell Rule Than Random Selection. In *ICML*, 2015.
- Nutini, Julie, Sepeshy, Behrooz, Laradji, Issam, Schmidt, Mark, Koepke, Hoyt, and Virani, Alim. Convergence Rates for Greedy Kaczmarz Algorithms, and Faster Randomized Kaczmarz Rules Using the Orthogonality Graph. In *UAI*, 2016.
- Qu, Zheng and Richtárik, Peter. Coordinate Descent with Arbitrary Sampling I: Algorithms and Complexity. *arXiv*, 1412.8060, 2014a.
- Qu, Zheng and Richtárik, Peter. Coordinate Descent with Arbitrary Sampling II: Expected Separable Overapproximation. *arXiv*, 1412.8063, 2014b.
- Qu, Zheng, Richtárik, Peter, and Zhang, Tong. Randomized Dual Coordinate Ascent with Arbitrary Sampling. In *NIPS*, 2015.
- Qu, Zheng, Richtárik, Peter, Takáč, Martin, and Fercoq, Olivier. SDNA: Stochastic Dual Newton Ascent for Empirical Risk Minimization. In *ICML*, 2016.
- Rahimi, Ali and Recht, Benjamin. Random Features for Large-Scale Kernel Machines. In *NIPS*, 2007.

- Recht, Benjamin and Ré, Christopher. Parallel Stochastic Gradient Algorithms for Large-Scale Matrix Completion. *Mathematical Programming Computation*, 5(2): 201–226, 2013.
- Richtárik, Peter and Takáč, Martin. Iteration Complexity of Randomized Block-Coordinate Descent Methods for Minimizing a Composite Function. *Mathematical Programming*, 114, 2014.
- Rudi, Alessandro, Carratino, Luigi, and Rosasco, Lorenzo. FALKON: An Optimal Large Scale Kernel Method. *arXiv*, 1705.10958, 2017.
- Sparks, Evan R., Venkataraman, Shivaram, Kaftan, Tomer, Franklin, Michael, and Recht, Benjamin. KeystoneML: Optimizing Pipelines for Large-Scale Advanced Analytics. In *ICDE*, 2017.
- Strohmer, Thomas and Vershynin, Roman. A Randomized Kaczmarz Algorithm with Exponential Convergence. *Journal of Fourier Analysis and Applications*, 15 (1), 2009.
- Tseng, Paul and Yun, Sangwoon. A Coordinate Gradient Descent Method for Nonsmooth Separable Minimization. *Mathematical Programming*, 117(1), 2009.
- Tu, Stephen, Roelofs, Rebecca, Venkataraman, Shivaram, and Recht, Benjamin. Large Scale Kernel Learning using Block Coordinate Descent. *arXiv*, 1602.05310, 2016.
- Tu, Stephen, Venkataraman, Shivaram, Wilson, Ashia C., Gittens, Alex, Jordan, Michael I., and Recht, Benjamin. Breaking Locality Accelerates Block Gauss-Seidel. *arXiv*, 1701.03863, 2017.
- Wilson, Ashia C., Recht, Benjamin, and Jordan, Michael I. A Lyapunov Analysis of Momentum Methods in Optimization. *arXiv*, 1611.02635, 2016.