Accelerating Greedy Coordinate Descent Methods

Haihao (Sean) Lu, Robert M. Freund, and Vahab Morrokni

MIT and Google Research

ISMP Bordeaux, July 2018

Paper

Conference paper:

"Accelerating Greedy Coordinate Descent Methods"

to be presented at ICML Stockholm July 2018

Literature on Coordinate Descent

Lots of excellent papers, here are some:

- Beck and Tetruashvili, *On the convergence of block coordinate descent type methods*
- Fercoq and Richtarik, *Accelerated, parallel, and proximal coordinate descent*
- Gurbuzbalaban, Ozdaglar, Parrilo,Vanli, *When cyclic coordinate descent outperforms randomized coordinate descent*
- Lee and Sidford, *Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems*
- Lin, Mairal, and Harchaoui, *A universal catalyst for first-order optimization*
- Locatello, Raj, Reddy, Rätsch, Schölkopf, Stich, Jaggi, *On matching pursuit and coordinate descent*
- Lu and Xiao, *On the complexity analysis of randomized block-coordinate descent methods*
- Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*
- Nutini, Schmidt, Laradji, Friedlander, and Koepke, *Coordinate descent converges faster with the Gauss-Southwell rule than random selection*
- Richtarik and Takac, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*
- Wilson, Recht, and Jordan, *A Lyapunov analysis of momentum methods in optimization*

Outline

- Accelerated Coordinate Descent Framework
- Accelerated Semi-Greedy Coordinate Descent (ASCD)
- ASCD under Strong Convexity
- Accelerated Greedy Coordinate Descent
- Numerical Experiments

Problem of Interest, and Coordinate-wise L-smoothness

$$P: \quad f^* := \quad \text{minimum}_x \quad f(x)$$

$$\text{s.t.} \quad x \in \mathbb{R}^n$$

where $f(\cdot)$ is a differentiable convex function

Coordinate-wise L-smoothness

$f(\cdot)$ is coordinate-wise L-smooth for the vector of parameters $L := (L_1, L_2, \ldots, L_n)$ if for all $x \in \mathbb{R}^n$ and $h \in \mathbb{R}$ it holds that:

$$|\nabla_i f(x + he_i) - \nabla_i f(x)| \leq L_i |h| \ , \ i = 1, \ldots, n \ ,$$

where $\nabla_i f(\cdot)$ denotes the i^{th} coordinate of $\nabla f(\cdot)$ and e_i is i^{th} unit coordinate vector, for $i = 1, \ldots, n$.

Coordinate-wise L-smoothness and $\mathbf{L}$ notation

Coordinate-wise L-smoothness

$f(\cdot)$ is coordinate-wise L-smooth for the vector of parameters $L := (L_1, L_2, \ldots, L_n)$ if for all $x \in \mathbb{R}^n$ and $h \in \mathbb{R}$ it holds that:

$$|\nabla_i f(x + he_i) - \nabla_i f(x)| \leq L_i |h| \ , \ \ i = 1, \ldots, n \ ,$$

where $\nabla_i f(\cdot)$ denotes the i^{th} coordinate of $\nabla f(\cdot)$ and e_i is i^{th} unit coordinate vector, for $i = 1, \ldots, n$.

Define the norm $\|x\|_L := \sqrt{\sum_{i=1}^n L_i x_i^2}$

and dual norm $\|v\|_{L^{-1}} := \sqrt{\sum_{i=1}^n L_i^{-1} v_i^2}$

Accelerated Coordinate Descent Framework

Accelerated Coordinate Descent Framework (without Strong Convexity)

Given $f(\cdot)$ with coordinate-wise smoothness parameter L, initial point x^0 and $z^0 := x^0$. Define step-size parameters $\theta_i \in (0, 1]$ recursively by $\theta_0 := 1$ and θ_{i+1} satisfies $\frac{1}{\theta_{i+1}^2} - \frac{1}{\theta_{i+1}} = \frac{1}{\theta_i^2}$.

For $k = 1, 2, \ldots,$ do:

Define $y^k := (1 - \theta_k)x^k + \theta_k z^k$

Choose coordinate j_k^1 (by some rule)

Compute $x^{k+1} := y^k - \frac{1}{L_{j_k^1}} \nabla_{j_k^1} f(y^k) e_{j_k^1}$

Choose coordinate j_k^2 (by some rule)

Compute $z^{k+1} := z^k - \frac{1}{nL_{j_k^2}\theta_k} \nabla_{j_k^2} f(y^k) e_{j_k^2}$.

Accelerated Randomized Coordinate Descent (ARCD)

Accelerated Randomized Coordinate Descent (ARCD) is the specification:

Accelerated Randomized Coordinate Descent (ARCD) (without Strong Convexity)

Given $f(\cdot)$ with coordinate-wise smoothness parameter L, initial point x^0 and $z^0 := x^0$. Define step-size parameters $\theta_i \in (0, 1]$ recursively by $\theta_0 := 1$ and θ_{i+1} satisfies $\frac{1}{\theta_{i+1}^2} - \frac{1}{\theta_{i+1}} = \frac{1}{\theta_i^2}$.

For $k = 1, 2, \ldots,$ do:

Define $y^k := (1 - \theta_k)x^k + \theta_k z^k$

Choose coordinate j_k^1 by $j_k^1 :\sim \mathcal{U}[1, \cdots, n]$

Compute $x^{k+1} := y^k - \frac{1}{L_{j_k^1}} \nabla_{j_k^1} f(y^k) e_{j_k^1}$

Choose coordinate j_k^2 by $j_k^2 = j_k^1$

Compute $z^{k+1} := z^k - \frac{1}{nL_{j_k^2}\theta_k} \nabla_{j_k^2} f(y^k) e_{j_k^2}$.

On Accelerated Randomized Coordinate Descent (ARCD)

- ARCD is well-studied
- ARCD updates 1 coordinate per iteration, hence x^k is k-sparse
- avoids computation of full gradient, which can save computation (or not) depending on the application
- randomization of x-update slows objective function improvement in practice
- Accelerated convergence guarantee (in expectation), for example [FR2015] :

$$E\left[f(x^k) - f(x^*)\right] \leq \frac{2n^2}{(k+1)^2}\|x^* - x^0\|_L^2 \ ,$$

where the expectation is on the random variables used to define the first k iterations

Accelerated Greedy Coordinate Descent (AGCD)

Accelerated Greedy Coordinate Descent (AGCD) is the specification:

Accelerated Greedy Coordinate Descent (AGCD) (without Strong Convexity)

Given $f(\cdot)$ with coordinate-wise smoothness parameter L, initial point x^0 and $z^0 := x^0$. Define step-size parameters $\theta_i \in (0, 1]$ recursively by $\theta_0 := 1$ and θ_{i+1} satisfies $\frac{1}{\theta_{i+1}^2} - \frac{1}{\theta_{i+1}} = \frac{1}{\theta_i^2}$.

For $k = 1, 2, \ldots,$ do:

Define $y^k := (1 - \theta_k)x^k + \theta_k z^k$

Choose coordinate j_k^1 by $j_k^1 := \arg\max_i \frac{1}{\sqrt{L_i}} |\nabla_i f(y^k)|$

Compute $x^{k+1} := y^k - \frac{1}{L_{j_k^1}} \nabla_{j_k^1} f(y^k) e_{j_k^1}$

Choose coordinate j_k^2 by $j_k^2 = j_k^1$

Compute $z^{k+1} := z^k - \frac{1}{nL_{j_k^2}\theta_k} \nabla_{j_k^2} f(y^k) e_{j_k^2}$.

On Accelerated Greedy Coordinate Descent (AGCD)

- AGCD has not been studied in the literature (that we are aware of)
- AGCD updates 1 coordinate per iteration, hence x^k is k-sparse
- AGCD computes the full gradient at each iteration, which can be expensive (or not) depending on the application
- the greedy nature of the x-update speeds convergence in practice
- no convergence results known for AGCD, in fact we suspect that there are examples where $O(1/k^2)$ convergence fails
- we observe $O(1/k^2)$ for AGCD in practice
- we will argue (later on) why $O(1/k^2)$ fails in theory
- we will also argue why $O(1/k^2)$ is observed in practice

Accelerated Semi-greedy Coordinate Descent (ASCD)

Accelerated Semi-greedy Coordinate Descent

(ASCD)

Accelerated Semi-greedy Coordinate Descent (ASCD)

Accelerated Semi-greedy Coordinate Descent (ASCD) is the specification:

Accelerated Semi-greedy Coordinate Descent (ASCD) (without Strong Convexity)

Given $f(\cdot)$ with coordinate-wise smoothness parameter L, initial point x^0 and $z^0 := x^0$. Define step-size parameters $\theta_i \in (0, 1]$ recursively by $\theta_0 := 1$ and θ_{i+1} satisfies $\frac{1}{\theta_{i+1}^2} - \frac{1}{\theta_{i+1}} = \frac{1}{\theta_i^2}$.

For $k = 1, 2, \ldots,$ do:

Define $y^k := (1 - \theta_k)x^k + \theta_k z^k$

Choose coordinate j_k^1 by $j_k^1 := \arg\max_i \frac{1}{\sqrt{L_i}} |\nabla_i f(y^k)|$

Compute $x^{k+1} := y^k - \frac{1}{L_{j_k^1}} \nabla_{j_k^1} f(y^k) e_{j_k^1}$

Choose coordinate j_k^2 by $j_k^2 :\sim \mathcal{U}[1, \cdots, n]$

Compute $z^{k+1} := z^k - \frac{1}{nL_{j_k^2}\theta_k} \nabla_{j_k^2} f(y^k) e_{j_k^2}$.

On Accelerated Semi-greedy Coordinate Descent (ASCD)

- ASCD and its complexity analysis is the new theoretical contribution of this paper
- ASCD updates 2 coordinates per iteration, hence x^k is $2k$-sparse
- computes the full gradient at each iteration, which can be expensive (or not) depending on the application
- the greedy nature of the x-update speeds convergence in practice
- Accelerated convergence guarantee on next slide ...

Computational Guarantee for Accelerated Semi-greedy Coordinate Descent (ASCD)

At each iteration k of ASCD the random variable j_k^2 is introduced, and therefore x^k depends on the realization of the random variable

$$\xi_k := \{j_0^2, \ldots, j_{k-1}^2\}$$

Theorem: Convergence Bound for Accelerated Semi-greedy Coordinate Descent (ASCD)

Consider the Accelerated Semi-Greedy Coordinate Descent algorithm. If $f(\cdot)$ is coordinate-wise L-smooth, it holds for all $k \geq 1$ that:

$$E_{\xi_k}\left[f(x^k) - f(x^*)\right] \leq \frac{n^2\theta_{k-1}^2}{2}\|x^* - x^0\|_L^2 \leq \frac{2n^2}{(k+1)^2}\|x^* - x^0\|_L^2 .$$

Accelerated Semi-greedy Coordinate Descent (ASCD) under Strong Convexity

Accelerated Semi-greedy Coordinate Descent

(ASCD)

under Strong Convexity

Accelerated Semi-greedy Coordinate Descent (ASCD) under Strong Convexity

We begin with the definition of strong convexity with respect to $\|\cdot\|_L$ due to [LX2015]:

μ-strong convexity with respect to $\|\cdot\|_L$

$f(\cdot)$ is μ-strongly convex with respect to $\|\cdot\|_L$ if for all $x, y \in \mathbb{R}^n$ it holds that:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|_L^2 \ .$$

Note that μ can be viewed as an extension of the condition number of $f(\cdot)$ in the traditional sense since μ is defined relative to the coordinate smoothness coefficients through $\|\cdot\|_L$

Accelerated Coordinate Descent Framework under Strong Convexity

Accelerated Coordinate Descent Framework (μ-strongly convex case)

Given $f(\cdot)$ with coordinate-wise smoothness parameter L and strong convexity parameter $\mu > 0$, initial point x^0 and $z^0 := x^0$. Define the parameters $a = \frac{\sqrt{\mu}}{n+\sqrt{\mu}}$ and $b = \frac{\mu a}{n^2}$.

For $k = 1, 2, \ldots,$ do:

Define $y^k := (1 - \theta_k)x^k + \theta_k z^k$

Choose coordinate j_k^1 (by some rule)

Compute $x^{k+1} := y^k - \frac{1}{L_{j_k^1}} \nabla_{j_k^1} f(y^k) e_{j_k^1}$

Compute $u^k := \frac{a^2}{a^2+b} z^k + \frac{b}{a^2+b} y^k$

Choose coordinate j_k^2 (by some rule)

Compute $z^{k+1} = u^k - \frac{a}{a^2+b} \frac{1}{nL_{j_k^2}} \nabla f_{j_k^2}(y^k) e_{j_k^2}$.

Accelerated Semi-Greedy Coordinate Descent under Strong Convexity

Accelerated Semi-greedy Coordinate Descent (μ-strongly convex case)

Given $f(\cdot)$ with coordinate-wise smoothness parameter L and strong convexity parameter $\mu > 0$, initial point x^0 and $z^0 := x^0$. Define the parameters $a = \frac{\sqrt{\mu}}{n+\sqrt{\mu}}$ and $b = \frac{\mu a}{n^2}$.

For $k = 1, 2, \ldots,$ do:

Define $y^k := (1 - \theta_k)x^k + \theta_k z^k$

Choose coordinate j_k^1 by $j_k^1 := \arg\max_i \frac{1}{\sqrt{L_i}} |\nabla_i f(y^k)|$

Compute $x^{k+1} := y^k - \frac{1}{L_{j_k^1}} \nabla_{j_k^1} f(y^k) e_{j_k^1}$

Compute $u^k := \frac{a^2}{a^2+b} z^k + \frac{b}{a^2+b} y^k$

Choose coordinate j_k^2 by $j_k^2 :\sim \mathcal{U}[1, \cdots, n]$

Compute $z^{k+1} = u^k - \frac{a}{a^2+b} \frac{1}{nL_{j_k^2}} \nabla f_{j_k^2}(y^k) e_{j_k^2}$.

Computational Guarantee for Accelerated Semi-greedy Coordinate Descent (ASCD) under Strong Convexity

Theorem: Convergence Bound for Accelerated Semi-greedy Coordinate Descent (ASCD) under Strong Convexity

Consider the Accelerated Semi-Greedy Coordinate Descent algorithm in the strongly convex case. If $f(\cdot)$ is coordinate-wise L-smooth and μ-strongly convex, it holds for all $k \geq 1$ that:

$$E_{\xi_k}\left[f(x^k) - f^* + \frac{n^2}{2}(a^2+b)\|z^k - x^*\|_L^2\right] \leq \left(1 - \frac{\sqrt{\mu}}{n+\sqrt{\mu}}\right)^k \left(f(x^0) - f^* + \frac{n^2}{2}(a^2+b)\|x^0 - x^*\|_L^2\right).$$

In particular, it holds that:

$$E_{\xi_k}\left[f(x^k) - f^*\right] \leq \left(1 - \frac{\sqrt{\mu}}{n+\sqrt{\mu}}\right)^k \left(f(x^0) - f^* + \frac{n^2}{2}(a^2+b)\|x^0 - x^*\|_L^2\right).$$

Observe that this is an accelerated linear convergence rate $\approx (1 - \sqrt{u}/n)$

Accelerated Greedy Coordinate Descent (AGCD)

Accelerated Greedy Coordinate Descent

(AGCD)

Accelerated Greedy Coordinate Descent (AGCD) (without Strong Convexity)

Accelerated Greedy Coordinate Descent (AGCD)

Given $f(\cdot)$ with coordinate-wise smoothness parameter L, initial point x^0 and $z^0 := x^0$. Define step-size parameters $\theta_i \in (0, 1]$ recursively by $\theta_0 := 1$ and θ_{i+1} satisfies $\frac{1}{\theta_{i+1}^2} - \frac{1}{\theta_{i+1}} = \frac{1}{\theta_i^2}$.

For $k = 1, 2, \ldots,$ do:

Define $y^k := (1 - \theta_k)x^k + \theta_k z^k$

Choose coordinate j_k^1 by $j_k^1 := \arg\max_i \frac{1}{\sqrt{L_i}} |\nabla_i f(y^k)|$

Compute $x^{k+1} := y^k - \frac{1}{L_{j_k^1}} \nabla_{j_k^1} f(y^k) e_{j_k^1}$

Choose coordinate j_k^2 by $j_k^2 = j_k^1$

Compute $z^{k+1} := z^k - \frac{1}{nL_{j_k^2}\theta_k} \nabla_{j_k^2} f(y^k) e_{j_k^2}$.

Why AGCD fails (in theory)

In the discrete-time setting, one can construct a Lyapunov energy function of the form:

$$E_k = A_k(f(x^k) - f^*) + \tfrac{1}{2}\|x^* - z^k\|_L^2$$

where A_k is a parameter sequence with $A_k \sim O(k^2)$.

Virtually all proof techniques for acceleration methods can be equivalently written as showing that E_k is non-increasing in k, thereby yielding:

$$f(x^k) - f^* \leq \frac{E_k}{A_k} \leq \frac{E_0}{A_k} = O\left(1/k^2\right)$$

Why AGCD fails (in theory), continued

$$E_k = A_k(f(x^k) - f^*) + \tfrac{1}{2}\|x^* - z^k\|_L^2$$

In AGCD the greedy coordinate j_k^1 is chosen to yield the greatest guaranteed decrease in $f(\cdot)$.

But one needs to prove a decrease in E_k, which is not the same as a decrease in $f(\cdot)$.

The coordinate j_k^1 is not necessarily the greedy coordinate for E_k due to the presence of the second term $\|x^* - z^k\|_L^2$.

This explains why the greedy coordinate can fail to decrease E_k, at least in theory.

Because x^* is not known when running AGCD, there does not seem to be any way to find the greedy descent coordinate for the energy function E_k.

Why AGCD fails (in theory), continued

$$E_k = A_k(f(x^k) - f^*) + \tfrac{1}{2}\|x^* - z^k\|_L^2$$

In ASCD:

- we use the greedy coordinate to perform the x-update (which corresponds to the best coordinate decrease for $f(\cdot)$)
- we choose a random coordinate to perform the z-update (which corresponds to the second term in the energy function)

This tackles the problem of dealing with the second term of the energy function.

A concurrent paper with similar notions: [LRRRSSJ2018]

Locatello, Raj, Reddy, Rätsch, Schölkopf, Stich, Jaggi, *On matching pursuit and coordinate descent*, ICML 2018

Develops computational theory for matching pursuit algorithms, which can be viewed as a generalized version of greedy coordinate descent where the directions do not need to be orthogonal

The paper also develops an accelerated version of the matching pursuit algorithms, which turns out to be equivalent to ASCD when the chosen directions are orthogonal

Both works use a decoupling of the coordinate update for the $\{x^k\}$ sequence (with a greedy rule) and the $\{z^k\}$ sequence (with a randomized rule)

[LRRRSSJ2018] is consistent with the argument here as to why one cannot accelerate greedy coordinate descent in general

How to make AGCD work (in theory)

Consider the following technical condition:

Technical Condition

There exists a positive constant γ and an iteration number K such that for all $k \geq K$ it holds that:

$$\frac{1}{n}\sum_{i=0}^{k}\frac{1}{\theta_i}\langle \nabla f(y^i), z^i - x^* \rangle \leq \sum_{i=0}^{k}\frac{\gamma}{\theta_i}\nabla_{j_i} f(y^i)(z^i_{j_i} - x^*_{j_i}) \ ,$$

where $j_i = \arg\max_i \frac{1}{\sqrt{L_i}}|\nabla_i f(y^k)|$ is the greedy coordinate at iteration i.

We will give some intuition on this in a couple of slides. But first ...

Computational Guarantee for Accelerated Greedy Coordinate Descent (AGCD) under the Technical Condition

Theorem: Convergence Bound for Accelerated Greedy Coordinate Descent (AGCD)

Consider the Accelerated Greedy Coordinate Descent algorithm. If $f(\cdot)$ is coordinate-wise L-smooth and satisfies the Technical Condition with constant $\gamma \leq 1$, then it holds for all $k \geq K$ that:

$$f(x^k) - f(x^*) \leq \frac{2n^2\gamma}{(k+1)^2}\|x^* - x^0\|_L^2 \ .$$

(The Technical Condition arises from a reverse engineering of the structure of the acceleration proof.)

Note that if $\gamma < 1$ (which we always observe in practice), then AGCD will have a better convergence guarantee than ASCD or ARCD.

Why the Technical Condition ought to hold in general

Technical Condition

There exists a positive constant γ and an iteration number K such that for all $k \geq K$ it holds that:

$$\frac{1}{n}\sum_{i=0}^{k}\frac{1}{\theta_i}\langle \nabla f(y^i), z^i - x^*\rangle \leq \sum_{i=0}^{k}\frac{\gamma}{\theta_i}\nabla_{j_i} f(y^i)(z^i_{j_i} - x^*_{j_i}) \ ,$$

where $j_i = \arg\max_i \frac{1}{\sqrt{L_i}}|\nabla_i f(y^k)|$ is the greedy coordinate at iteration i.

The three sequence $\{x^k\}$, $\{y^k\}$ and $\{z^k\}$ ought to all converge to x^* .

Thus we can instead consider the inner product $\langle \nabla f(y^i), y^i - x^*\rangle$

For any j we have $|y^i_j - x^*_j| \geq \frac{1}{L_j}|\nabla_j f(y^i)|$, and therefore $|\nabla_j f(y^i)\cdot(y^i_j - x^*_j)| \geq \frac{1}{L_j}|\nabla_j f(y^i)|^2$.

The greedy coordinate is chosen by $j_i := \arg\max_j \frac{1}{L_j}|\nabla_j f(y^i)|^2$

It is reasonably likely that in most cases the greedy coordinate will yield a better product than the average of the components of the inner product.

Numerical Experiments

Numerical Experiments

Numerical Experiments

Linear Regression Problems

- least squares minimization: $\min_\beta \frac{1}{2n}\|y - X\beta\|_2^2$
- synthetic instances in order to control condition number $\kappa(X^T X)$
- $n = 200$, $p = 100$

Logistic Regression Problems

- logistic loss minimization: $\min_\beta \frac{1}{n}\sum_{i=1}^n \ln(1 + \exp(-y_i \beta^T x_i))$
- real problem instances taken from LIBSVM
- locally strongly convex with parameter μ, we assigned parameter $\bar{\mu}$ in the experiments

Linear Regression Experiments

Linear Regression Experiments

Prototypical Comparison of ARCD, ASCD, and AGCD on Linear Regression Problems

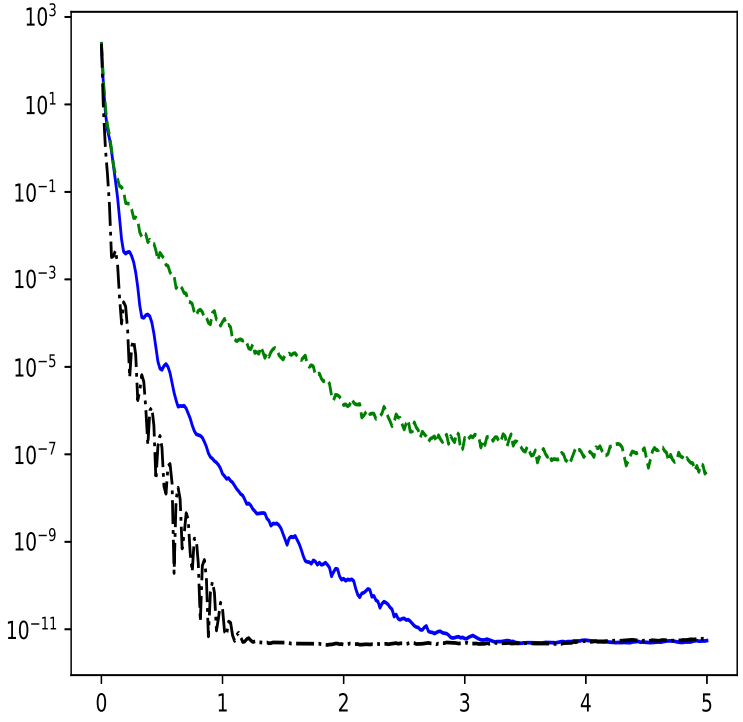
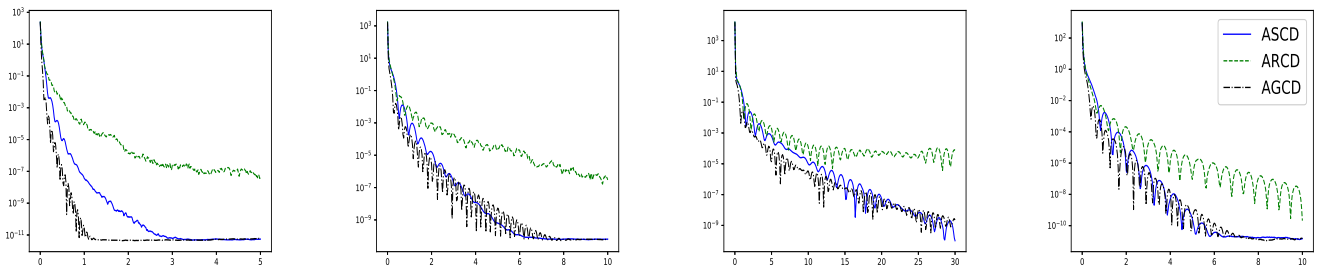
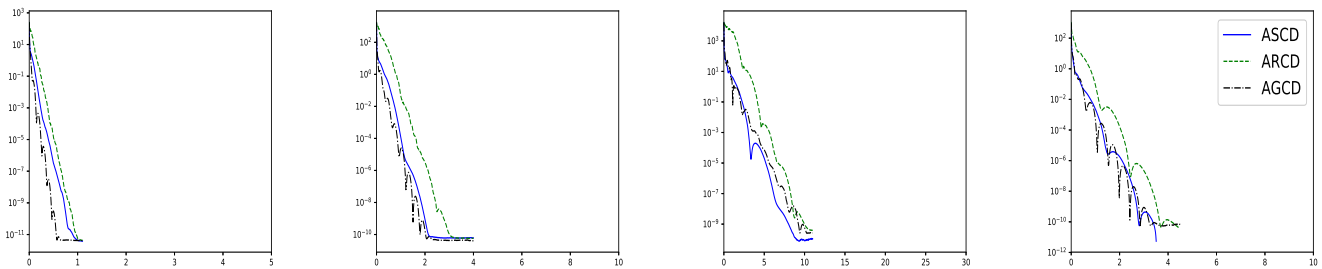


Figure: Plot showing the optimality gap versus run-time (in seconds) for a synthetic linear regression instance solved by **ARCD**, **ASCD**, **AGCD**.

Comparing the Methods on Linear Regression Problems with Different Conditions Numbers $\kappa(X^T X)$

	$\kappa = 10^2$	$\kappa = 10^3$	$\kappa = 10^4$	$\kappa = \infty$
Algorithm Framework 1 (non-strongly convex)				
Algorithm Framework 2 (strongly convex)				

Plots showing the optimality gap versus run-time (in seconds) for synthetic linear regression problems solved by **ARCD**, **ASCD**, **AGCD**.

Logisitic Regression Experiments

Logistic Regression Experiments

Prototypical Comparison of ARCD, ASCD, and AGCD on Logistic Regression Problems

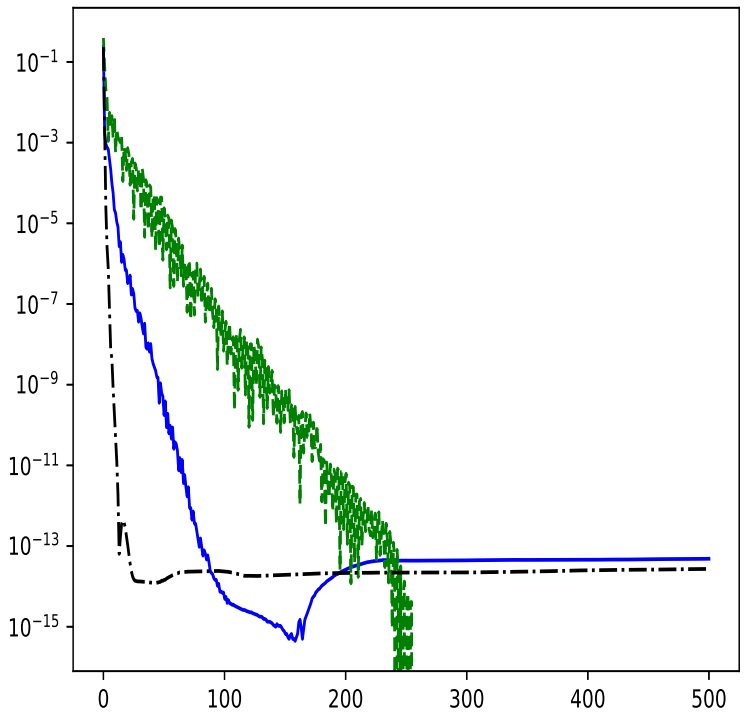
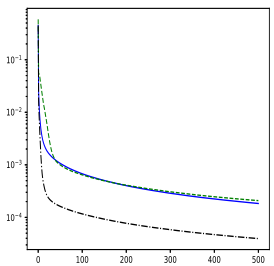
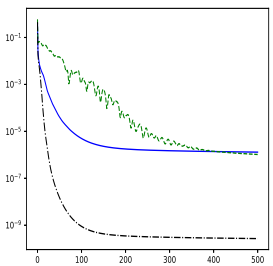
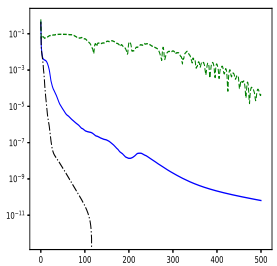
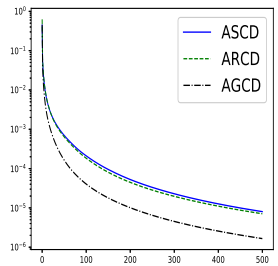
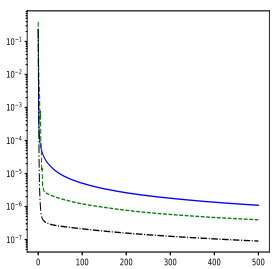
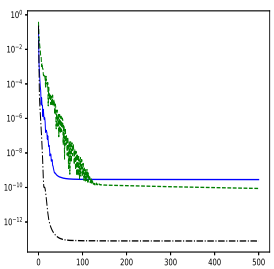
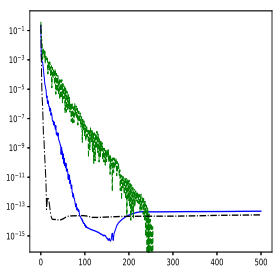
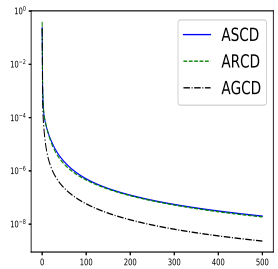


Figure: Plot showing the optimality gap versus run-time (in seconds) for the logistic regression instance **a1a** solved by **ARCD**, **ASCD**, **AGCD**.

Comparing the Methods on Logistic Regression Problems with Different Assigned Strong Convexity Parameters $\bar{\mu}$

Dataset	$\bar{\mu} = 10^{-3}$	$\bar{\mu} = 10^{-5}$	$\bar{\mu} = 10^{-7}$	$\bar{\mu} = 0$
w1a				
a1a				

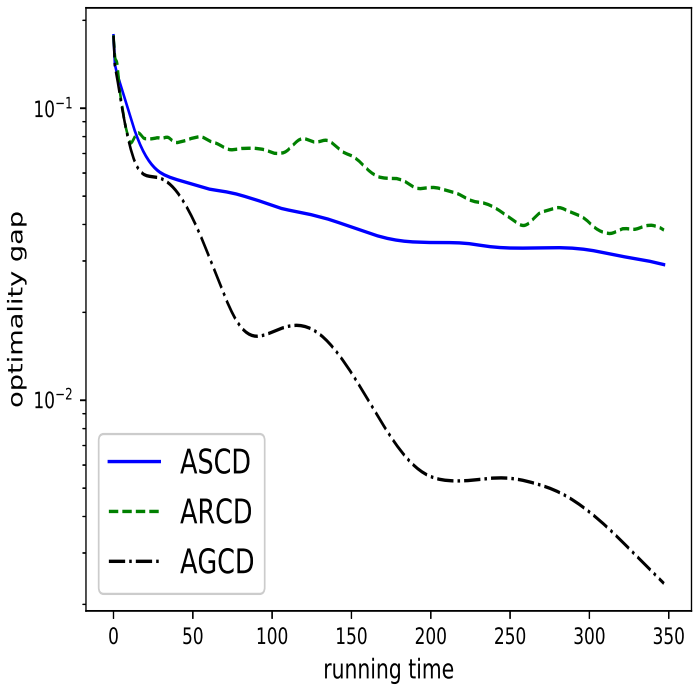
Plots showing the optimality gap versus run-time (in seconds) for the logistic regression instances w1a and a1a in LIBSVM, solved by **ARCD**, **ASCD**, **AGCD**.

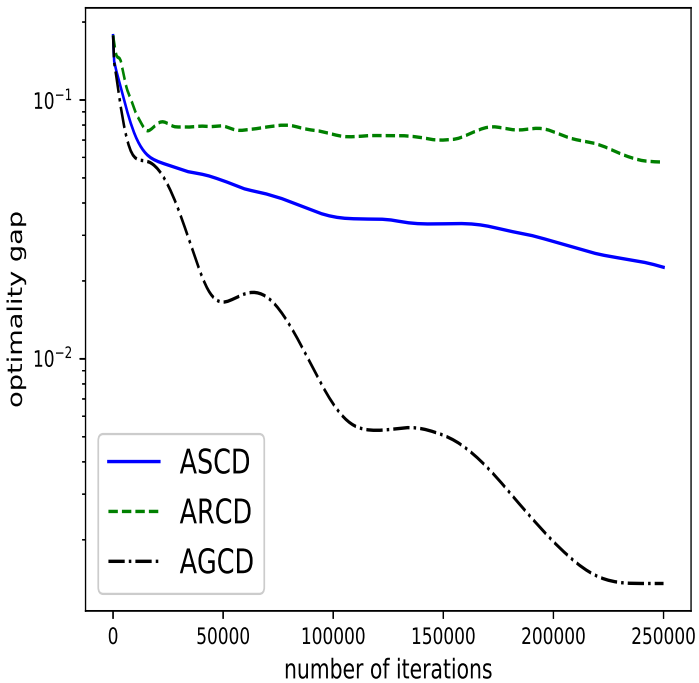
Empirical Values of γ arising from the Technical Condition

Dataset	γ
w1a	0.25
a1a	0.17
heart	0.413
madelon	0.24
rcv1	0.016

Largest observed values of γ for five different datasets in LIBSVM for $k \geq \bar{K} := 5000$.

Comparing the Algorithms using Running Time and the Number of Iterations





Plots showing the optimality gap versus run-time (in seconds) on the left and versus the number of iterations on the right, for the logistic regression instance **madelon**, solved by **ARCD**, **ASCD**, **AGCD.**

Conclusions/Remarks

AGCD:

- the natural accelerated version of Greedy Coordinate Descent
- unlikely that AGCD has an acceleration guarantee ($O(1/k^2)$)
- exhibits acceleration in practice
- extremely effective in practice
- Technical Condition "explains" acceleration in practice

ASCD:

- new theoretical contribution of this paper
- combines salient features of AGCD and ARCD
- acceleration guarantee ($O(1/k^2)$)
- accelerated linear convergence with rate $\approx (1 - \sqrt{\mu}/n)$ in strongly convex case
- very effective in practice

We thank Martin Jaggi as well as three excellent anonymous referees