ISMP 2018 - Bordeaux

Approximate Composite Minimization: Convergence Rates and Examples

S. Praneeth Karimireddy, **Sebastian U. Stich**, Martin Jaggi

MLO Lab, EPFL, Switzerland
sebastian.stich@epfl.ch

July 4, 2018

[AISTATS 2018] Adaptive balancing of gradient and update computation times

Convex Optimization

Black-Box Optimization

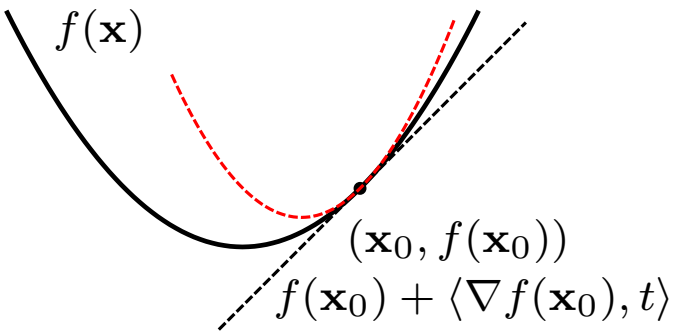
$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \qquad \mathbf{x} \rightarrow \boxed{\phantom{f(\mathbf{x})}} \rightarrow f(\mathbf{x})$$

- $f\colon \mathbb{R}^n \to \mathbb{R}$, assume $f \in C^1$
- oracle access to f $(f(\mathbf{x}), \nabla f(\mathbf{x}))$ or just $(\nabla_i f(\mathbf{x}))$
- Approximate solution $\mathbf{x}_\epsilon$: $f(\mathbf{x}_\epsilon) - \min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \leq \epsilon$
- $N(f, \mathcal{A}, \epsilon)$ number of oracle calls for algorithm $\mathcal{A}$ to find an approximate solution
- Worst case complexity of algorithm $\mathcal{A}$ on a class $\mathcal{F}$: $N(\mathcal{A}, \epsilon) = \max_{f\in\mathcal{F}} N(f, \mathcal{A}, \epsilon)$

Function Classes

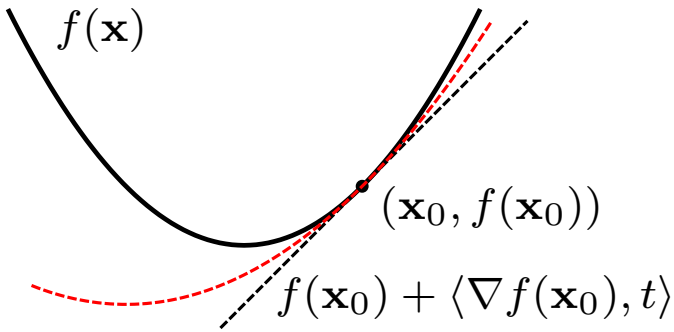
Upper bound: $f \in C^1_{\mathbf{L}}$

$$f(\mathbf{y}) \leq \underbrace{f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2_{\mathbf{L}}}_{=: u^{\mathbf{L}}_{\mathbf{x}}(\mathbf{y})}$$



Lower bound: $f \in C_\mu$

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



Consequences:

- $\mathbf{x}^\star := \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x})$ unique
- $f(\mathbf{x}) - f(\mathbf{x}^\star) \leq \frac{1}{2\mu} \|\nabla f(\mathbf{x})\|^2_{\mathbf{L}^{-1}}$
- condition number $\kappa := \|\mathbf{L}\| / \mu$

$$\|\mathbf{x}\|_{\mathbf{L}} := \langle \mathbf{x}, \mathbf{L}\mathbf{x} \rangle$$

$$\mu\mathbf{L} \preceq \nabla^2 f(\mathbf{x}) \preceq \mathbf{L}$$

general framework

$$F(\mathbf{x}) := f(\mathbf{x}) + \Psi(\mathbf{x})$$

Setting:

- $f \in C^1_{\mathbf{L}}$, $f \in C^1_{\mu_f}$ for $\mu_f \geq 0$.
- $\Psi \in C_{\mu_\Psi}$ for $\mu_\Psi \geq 0$.

Goal:

$$\min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) := f(\mathbf{x}) + \Psi(\mathbf{x})$$

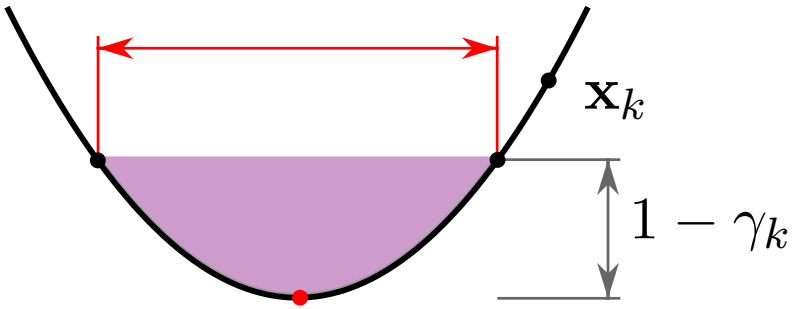
Method 0: $\mathbf{x}_0 \in \mathbb{R}^n$

$$\mathbf{x}^\star = \min_{\mathbf{u} \in \mathbb{R}^n} F(\mathbf{x}_0 + \mathbf{u})$$

This problem is too hard! Define easier subproblems!

Relaxing Accuracy:

Introduce relative accuracy parameter $\gamma_k \geq 0$:



Method 1: $\mathbf{x}_0 \in \mathbb{R}^n$

$$F(\mathbf{x}_{k+1}) \leq (1-\gamma_k)\underbrace{F(\mathbf{x}_k)}_{\text{old value}} + \gamma_k \underbrace{\min_{\mathbf{u}\in\mathbb{R}^n} F(\mathbf{x}_k+\mathbf{u})}_{\text{best value}}$$

Verifying the condition requires function evaluations, which might be elusive!

Option II: Optimize the upper bound

Upper bound:

$$F(\mathbf{x}_k + \mathbf{u}) \leq \underbrace{u^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_k + \mathbf{u}) + \Psi(\mathbf{x}_k + \mathbf{u})}_{=:m^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_k+\mathbf{u})}$$

Note: It suffices to pick $\mathbf{L}$ s.t. $F(\mathbf{x}_k + \mathbf{u}) \leq m^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_k + \mathbf{u})$, i.e. $f(\mathbf{x}_k + \mathbf{u}) \leq u^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_k + \mathbf{u})$ is not required.

Method 2: $\mathbf{x}_0 \in \mathbb{R}^n$

$$F(\mathbf{x}_{k+1}) \leq m^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_{k+1}) \leq (1-\gamma_k)\underbrace{F(\mathbf{x}_k)}_{\text{old value}} + \gamma_k \underbrace{\min_{\mathbf{u}\in\mathbb{R}^n} m^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_k + \mathbf{u})}_{\text{best value}}$$

This requires computation of $\nabla f(\mathbf{x}_k)$ (to evaluate $u^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_k + \mathbf{u})$) and $\Psi(\mathbf{x}_k + \mathbf{u})$ instead of $F(\mathbf{x}_k + \mathbf{u})$.

Complexity Results

$$N(\textbf{Method 2}, \epsilon) = \min \left\{ \underbrace{\frac{2D^2}{\bar{\gamma}\epsilon}}_{\text{smooth case}}, \underbrace{\frac{1+\mu_\Psi}{(\mu_f+\mu_\Psi)\bar{\gamma}} \ln\left(\frac{F(\mathbf{x}_0)-F(\mathbf{x}^\star)}{\epsilon}\right)}_{\text{strongly convex case}} \right\}$$

$D = \max_{\mathbf{y}:\, F(\mathbf{y}) \leq F(\mathbf{x}_0)} \|\mathbf{y}-\mathbf{x}^\star\|_{\mathbf{L}}$, average accuracy $\bar{\gamma} = \frac{1}{k}\sum_{i=1}^k \gamma_k$.

- Both μ_f and μ_Ψ appear in the rate, thus $\mathbf{L}$ should ideally approximate the curvature of *both* f an Ψ.
- Extends and unifies **[Stich et al. 13], [Qu et al. 15], [Tappenden et al. 16], [Mutny, Richtárik, 18], [...]**

Examples

Gradient Method

$$\mathbf{L} = L\mathbf{I}_n$$

$$\mathbf{x}_{k+1} = \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^n} \left\{ f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{u} - \mathbf{x}_k \rangle + \frac{L}{2} \left\| \mathbf{u} - \mathbf{x}_k \right\|^2 + \Psi(\mathbf{x}_k + \mathbf{u}) \right\}$$

$\mathbf{L}^{-1}$ hard to compute: use an (arbitrary) iterative method to solve

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \gamma_k \operatorname*{argmin}_{\mathbf{u}\in\mathbb{R}^n} \left\{ \langle \nabla f(\mathbf{x}_k), \mathbf{u} - \mathbf{x}_k \rangle + \frac{1}{2} \left\| \mathbf{u} - \mathbf{x}_k \right\|_{\mathbf{L}}^2 + \Psi(\mathbf{x}_k + \mathbf{u}) \right\}$$

Requires full $\nabla f(\mathbf{x}_k)$, which might be elusive for $n \gg 1$.

Random Methods

$$T(\text{iteration}) \ll T(\nabla f)$$

Random Pursuit

Choose a (random) sketch matrix $\mathbf{U}_k \in \mathbb{R}^{n\times m}$, $\operatorname{rank}\mathbf{U}_k = m$.

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \gamma_k \underbrace{\operatorname*{argmin}_{\mathbf{u}\in span\,\mathbf{U}_k} \left\{ \langle \nabla f(\mathbf{x}_k), \mathbf{u}\rangle + \frac{1}{2}\|\mathbf{u}\|^2_{\mathbf{L}} + \Psi(\mathbf{x}_k+\mathbf{u}) \right\}}_{=:\ \operatorname*{argmin}_{\mathbf{u}\in R^n} m^{\mathbf{L}}_{\mathbf{x}_k|\mathbf{U}_k}(\mathbf{x}_k+\mathbf{u})}$$

Requires computation of $\mathbf{U}_k^\top \nabla f(\mathbf{x}_k)$.

Example: (Block)-Coordinate Descent

- $\mathbf{U}_k^\top \mathbf{U}_k = \mathbf{I}_m$, $\mathbf{U}_k^\top \nabla f(\mathbf{x}_k) = [\nabla_{i_1} f(\mathbf{x}_k), \dots, \nabla_{i_m} f(\mathbf{x}_k)]^\top$.
- Assume Ψ (block)-separable structure:
  $\Psi(\mathbf{x}_k + \mathbf{U}_k\mathbf{z}) = \Psi_{|\mathbf{U}_k^\perp}(\mathbf{x}_k) + \Psi_{|\mathbf{U}_k}(\mathbf{x}_k + \mathbf{U}_k\mathbf{z})$

How does this fit in the framework?

Option I: The γ_k-approximate solution on $m^{\mathbf{L}}_{\mathbf{x}_k|\mathbf{U}_k}$ can be seen as a γ'_k-approximate solution on $m^{\mathbf{L}}_{\mathbf{x}_k}$, $\gamma'_k \leq \gamma_k$.

Option II: For certain distributions of $\mathbf{U}_k$ the results can be explicitly stated.
Example:
If $\mathbb{E}\,\mathbf{U}_k\mathbf{U}_k^\top = \frac{m}{n}\mathbf{I}_n$ the complexity increases by a factor of $\frac{n}{m}$.

[KSJ18] also states explicit results for parallel updates.

Application: (Block) Coordinate Descent

$$[\nabla_{i_1} f(\mathbf{x}), \ldots, \nabla_{i_m} f(\mathbf{x})]$$

Block Coordinate Descent: sketch matrix $\mathbf{U}_k$, $\mathbf{U}_k^\top \mathbf{U}_k = \mathbf{I}_m$

m ($\ll n$)-dimensional subproblem:

$$m^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_{k+1}) \leq (1-\gamma_k) F(\mathbf{x}_k) + \gamma_k \min_{\mathbf{u} \in span\, \mathbf{U_k}} m^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_k + \mathbf{u})$$

Iteration comprises two steps:

1. Compute $\mathbf{U}_k^\top \nabla f(\mathbf{x}_k) = [\nabla_{i_1} f(\mathbf{x}_k), \dots \nabla_{i_m} f(\mathbf{x}_k)]^\top$
2. Compute approx. minimizer $\mathbf{x}_{k+1}$

Two different costs! Examples:

- Cache misses, expensive operations to compute $\nabla_i f(\mathbf{x}_k)$
- Hard subproblem, $\Psi(\mathbf{x}) = \|\mathbf{x}\|_{TV}$

Towards optimal balancing

Goal: Set accuracy γ_k to optimally 'balance' the costs

Feasible strategy:

- Suppose we use an interative algorithm to approximately solve the model $m^{\mathbf{L}}_{\mathbf{x}_k}$. I.e. $\mathbf{x}_k = \mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_t, \ldots, \mathbf{y}_T = \mathbf{x}_{k+1}$
- Each intermediate solution $\mathbf{y}_t$ corresponds to an approximate solution with parameter γ_k^t:

$$m^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{y}_t) \leq (1-\gamma_k^t)F(\mathbf{x}_k) + \gamma_k^t \min_{\mathbf{u} \in span\, \mathbf{U_k}} m^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_k + \mathbf{u})$$

- Thus we can express the progress in terms of iterations on the model:

$$p_k(t) := \underbrace{\gamma_k^t}_{\text{relative progress}} \underbrace{\left(F(\mathbf{x}_k) - \min_{\mathbf{u} \in span\, \mathbf{U_k}} m^{\mathbf{L}}_{\mathbf{x}_k}(\mathbf{x}_k + \mathbf{u}) \right)}_{\text{scale}}$$

Optimal balancing

- Each (inner) iteration $(\mathbf{y}_t \to \mathbf{y}_{t+1})$ denotes one 'unit' of time.
- Let $c_k = T(\mathbf{U_k}^\top \nabla f(\mathbf{x}_k))$ denote the time to compute the required coordinates of the gradient.

Optimal number of inner iterations $t^\star$:

$$t_k^\star := \operatorname*{argmax}_t \underbrace{\frac{p_k(t)}{t + c_k}}_{\text{progress per time spent}}$$

Can we compute $t_k^\star$? Can we ~~compute~~ approximate $t_k^\star$?
Predict t_k based on observations at iteration $k-1$. **Goal:** $t_k \approx t_k^\star$

(Constant) strategies for approximating $t_k^\star$

Constant Strategies:

one: Set $t_k \equiv 1$, $\forall k \geq 0$. — arbitrary constant

comp: $t_k = c_k$, 'balance' *time* — "best practice"

Can we measure $p_k(t_k)$? $\quad p_k(t_k) = F(\mathbf{x}_k) - m_{\mathbf{x}_k}^{\mathbf{L}}(\mathbf{x}_{k+1})$

And $p_k'(t_k)$?

$$\frac{d}{dt}\underbrace{\frac{p_k(t_k)}{t + c_k}}_{\text{progress per time spent}} = \frac{p_k'(t_k)(t_k + c_k) - p_k(t_k)}{(t_k + c_k)^2} =: g_k(t_k)$$

Can be estimated using $p_k'(t_k) \approx p_k(t_k) - p_k(t_{k-1})$.

(Adaptive) strategies for approximating $t_k^\star$

General adaptive strategy: Initialize $t_0 := 1$

Each iteration:

1. Estimate $g_k(t_k)$
2. $t_{k+1} = \mathcal{A}(t_k, g_k)$

Example strategy: **add**

$$t_{k+1} = \begin{cases} t_k + 1 & \text{if } g_k(t_k) > 0 \\ \max\{t_k - 1, 1\} & \text{otherwise} \end{cases}$$

Two more adaptive strategies:

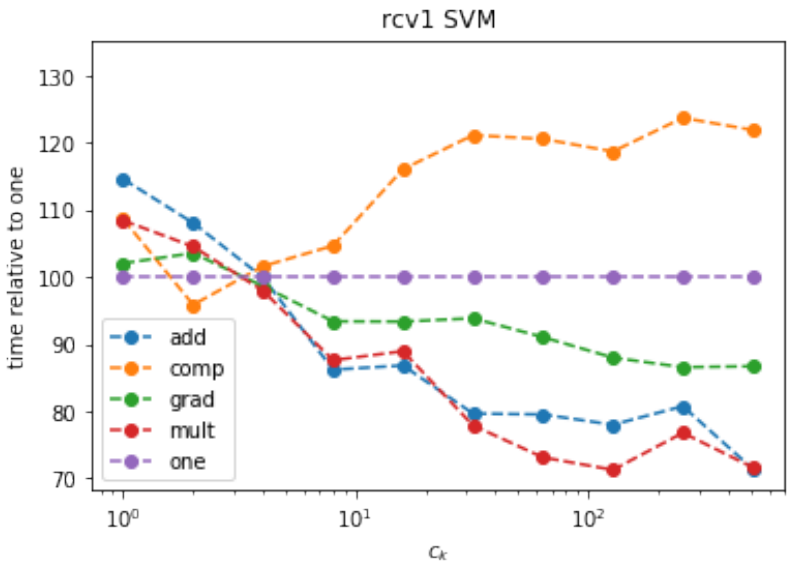
mult: $t_{k+1} = \{2t_k, \text{if } g_k(t_k) > 0, \max\{t_k - 1, 1\}, \text{else}$

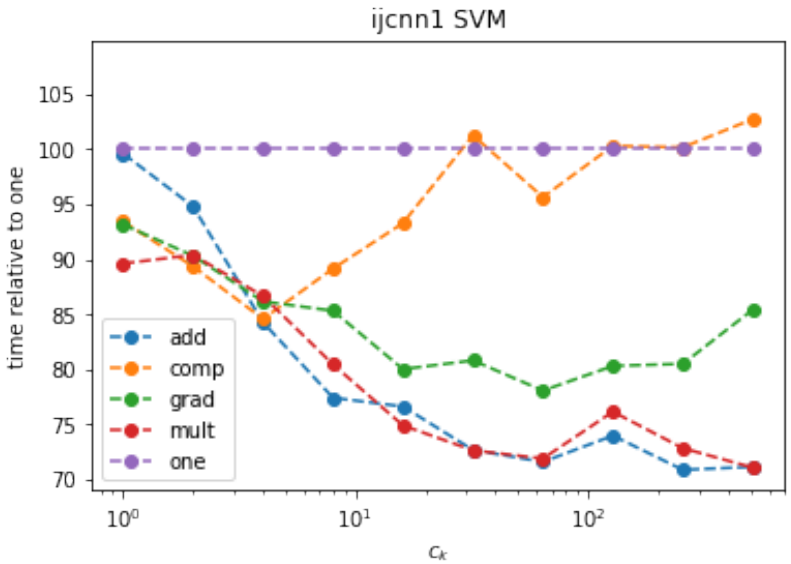
grad: $t_{k+1} = t_k + g_t$

Experiments

Improvement in Time

Time relative to constant strategy one:

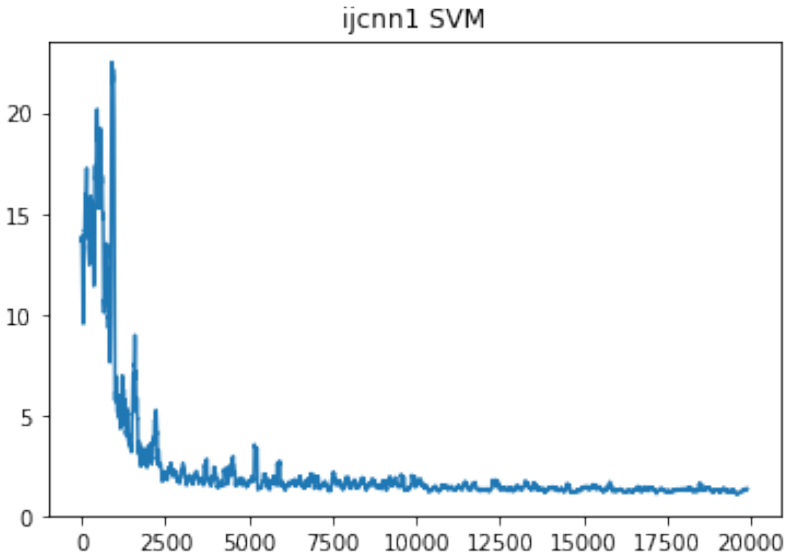




- **adaptivity** is important! precise rule less important
- "best practice" **comp** is not performing well!

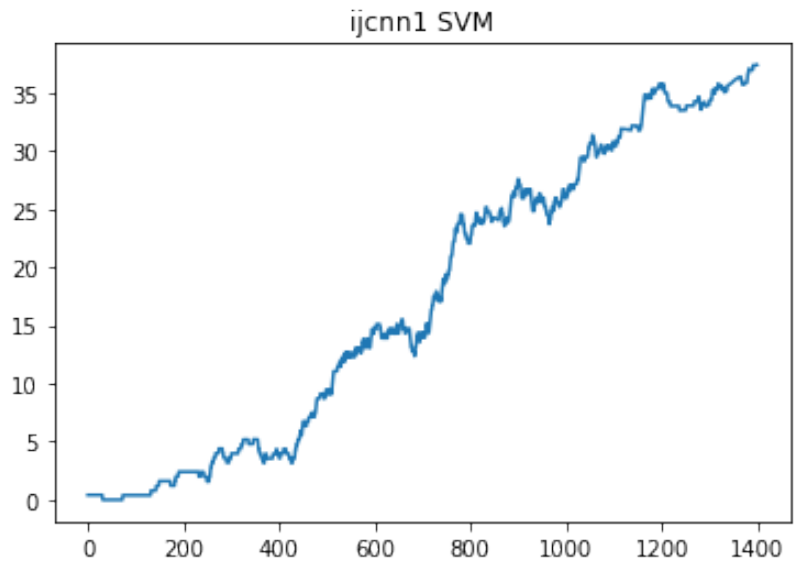
A closer look

mult strategy:



t_k values v.s. iteration k

Problem difficulty: (simulated)



t_k required to reach accuracy $\gamma_k \equiv 0.1$ vs. iteration k

less time is spent as the subproblems get harder

Conclusion

Contributions & Open Problem

Theory: We present a sound theoretical framework

- where subproblems need to be solved just approximately (with arbitrary iterative solver)
- convergence rate depends on average quality of approximation
- parallel, distributed and primal-dual extensions

Practice: We observe

- **adaptivity** is important, not all subproblems are the same!

Open problem: Proof for adaptive schemes missing!

- The proofs do not extend to the fully-adaptive setting, i.e. when γ_k depends on $\mathbf{x}_{k+1}$ (as it is the case for the adaptive strategies).

References

Stich et al. 13 S.U. Stich, C.L. Müller, B. Gärtner. Optimization of convex functions with Random Pursuit, SIAM J.Opt. 2013.

Qu et al. 15 Z. Qu, P. Richtárik, M. Takac, O. Fercoq. SDNA: Stochastic Dual Newton Ascent for Empirical Risk Minimization, 2015.

Tappenden et al. 16 R. Tappenden, P. Richtárik, J. Gondzio. Inexact Coordinate Descent: Complexity and Preconditioning, J. Opt. T& A, 2016.

Mutny, Richtárik, 18 M. Mutny, P. Richtárik, Parallel Stochastic Newton Method, J. Comp. Math, 2018.

KSJ18 S.P.R. Karimireddy, S.U. Stich, M. Jaggi, Adaptive balancing of gradient and update computation times using global geometry and approximate subproblems, PMLR 84, 2018.