Let's Make Block Coordinate Descent Go Fast!

Julie Nutini (UBC), Issam Laradji (UBC), Mark Schmidt (UBC) and Warren Hare (UBC Okanagan)

OVERVIEW:

Motivation:

- Block coordinate descent (BCD) methods are key tools in large-scale optimization.
  - Easy to implement.
  - Low memory requirements.
  - Cheap iteration costs.
  - Adaptability to distributed settings.
- Used for almost two decades to solve LASSO and SVMs.

→ **Any** improvements on convergence will affect many applications.

This work:

New greedy block selection rules: → Exploit structure to ensure more progress than classic Gauss-Southwell.

New higher-order update: → Update large blocks using message passing in graph-structured problems.

$$x_{b_k}^{k+1} = x_{b_k}^k + d^k$$

Active-set identification results: → Analysis for active-set identification and bounds on active-set complexity.

Approximate greedy block selection rules: → Tractable for variable block selection.

Greedy Block Selection Rules

- Consider the basic convex optimization problem:

$$\underset{x \in \mathbb{R}^n}{\text{argmin}} \; f(x),$$

where f is differentiable and n is large.

Block Gauss-Southwell (GS)

- Classic Gauss-Southwell rule selects block whose gradient has largest Euclidean norm,

$$\underset{b \in \mathcal{B}}{\text{argmax}} \, \|\nabla_b f(x^k)\|.$$

→ Tends to make more progress than cyclic or random block selection.

Block Gauss-Southwell-Lipschitz (GSL)

- Assume f is L_b-block-wise Lipschitz continuous,

$$\|\nabla_b f(x + U_b d) - \nabla_b f(x)\| \leq L_b \|d\|, \quad \text{for all } d.$$

- Minimizing quadratic bound from block-wise Lipschitz continuity, we have

$$b_k \in \underset{b \in \mathcal{B}}{\text{argmax}} \frac{\|\nabla_b f(x^k)\|_2}{\sqrt{L_b}}.$$

→ By using Lipschitz constants, guarantees more progress than GS rule.

Block Gauss-Southwell-Quadratic (GSQ)

- Obtain a better bound by measuring Lipschitz continuity using quadratic norms,

$$\|\nabla_b f(x + U_b d) - \nabla_b f(x)\|_{H_b^{-1}} \leq \|d\|_{H_b} = \sqrt{d^T H_b d},$$

where H_b is a global upper bound on the Hessian $\nabla^2_{bb} f$.

- Minimizing the Lipschitz quadratic bound measured in the quadratic norm, we have

$$b_k \in \underset{b \in \mathcal{B}}{\text{argmax}} \left\{ \|\nabla_b f(x^k)\|_{H_b^{-1}} \right\} \equiv \underset{b \in \mathcal{B}}{\text{argmax}} \left\{ \nabla_b f(x^k)^T H_b^{-1} \nabla_b f(x^k) \right\}.$$

→ Equivalent to the maximum improvement rule for quadratics.

→ May be difficult to find Hessian bounds H_b, depends on how we define blocks.

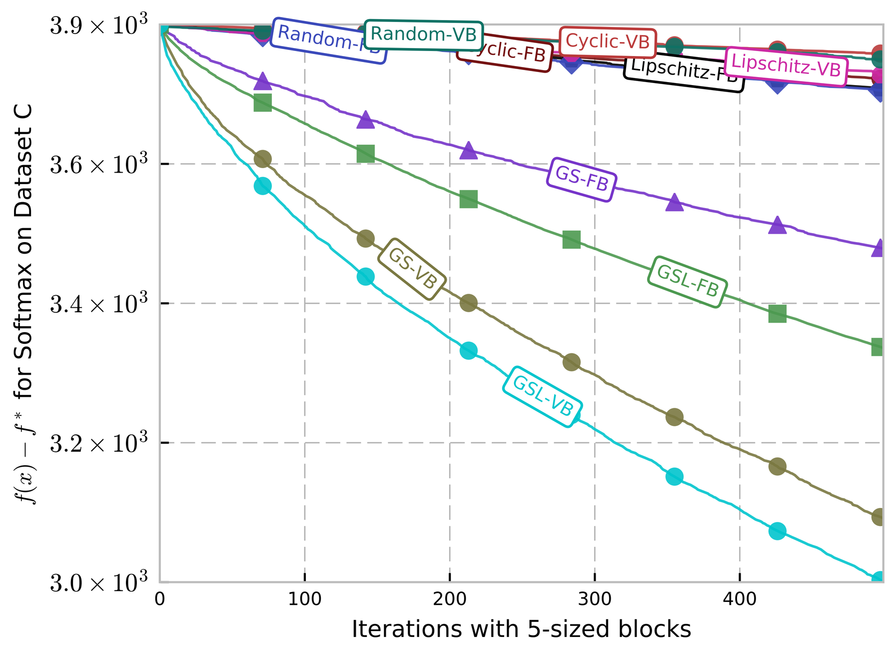
Block Gauss-Southwell-Diagonal (GSD)

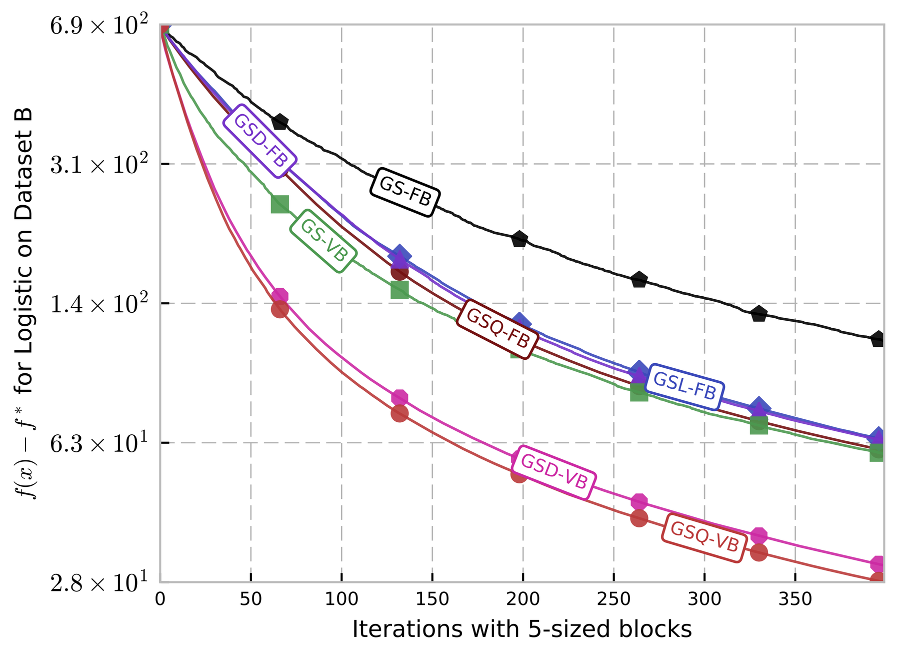
- By upper bounding the block-Hessian $\nabla^2_{bb} f$ by a diagonal matrix D_b, we have,

$$b_k \in \underset{b \in \mathcal{B}}{\text{argmax}} \left\{ \|\nabla_b f(x^k)\|_{D_b^{-1}} \right\} \equiv \underset{b \in \mathcal{B}}{\text{argmax}} \left\{ \sum_{i \in b} \frac{|\nabla_i f(x^k)|^2}{D_{b,i}} \right\}.$$

→ $D_{b,i}$ refer to the diagonal element corresponding to coordinate i in block b.

→ Guarantees more progress than GSL rule, may be easier to implement than GSQ rule.





→ We can define a set of possible blocks $\mathcal{B}$ using two strategies:

Fixed (FB): Initialize partition of variables into groups.

→ E.g., sort according to Lipschitz constants.

Variable (VB): Choose "best" M variables at each step.

→ Guarantees more progress, but GSL/GSQ seem intractable for variable blocks.

- ⋆ Approximate GSQ using $H_b = M_{b,b}$, where M is a fixed upper bounding matrix.
- ⋆ We can approximate GSD by assuming $D_{b,i} = d_i$ (same across all blocks b).
- ⋆ Approximate GSL using GSD rule with $D_{b,i} = \sum_j M_{i,j}$ (approximates L_b).

Message-Passing for Higher-Order Updates

- Consider a basic quadratic minimization problem restricted to the coordinates of block b,

$$\underset{x_b}{\text{argmin}} \; \frac{1}{2} x_b^T A_{bb} x_b - \tilde{c}^T x_b,$$

where

- $A_{bb} \in \mathbb{R}^{|b| \times |b|}$ is the submatrix of positive-definite and sparse matrix A
- $\tilde{c} = c_b - A_{b\bar{b}} x_{\bar{b}}$ is a vector with $\bar{b}$ defined as the complement of b

⋆ Solve this problem to find Newton updates and (for sparse quadratics) optimal updates.

- Using matrix factorization methods this costs $O(|b|^3)$.
- We exploit connection to Gaussian Markov random fields:

→ Update tree-structured blocks in $O(|b|)$ using Gaussian belief propagation.

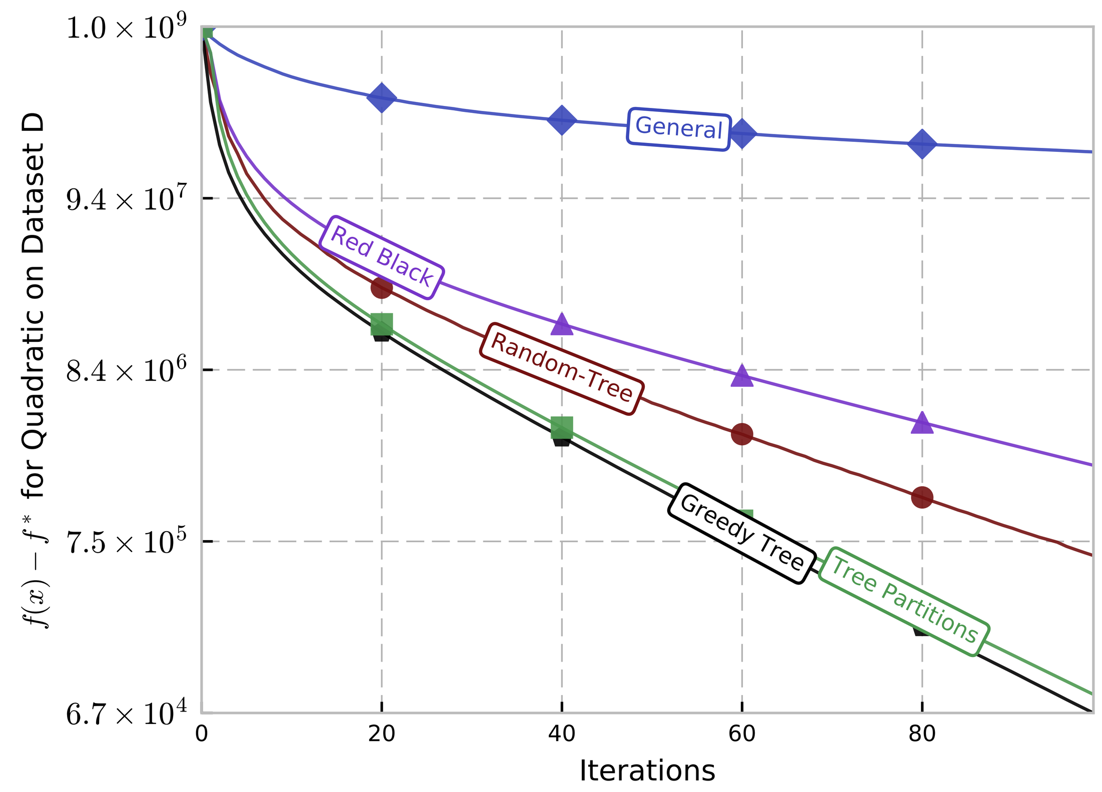
→ Equivalent to Gaussian elimination on tree-structured blocks.

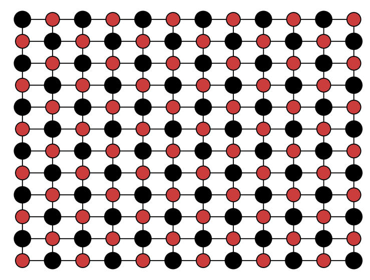
- Options for structured partitioning of the blocks into trees:

(a) **Classic red-black** → loses dependencies.

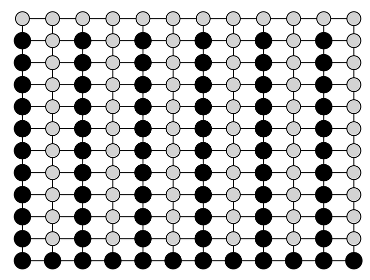
(b) **Fixed tree partition** → maintains dependencies, $|b| = n/2$.

(c) **Variable greedy tree** → spans all corners of graph, $|b| \approx 2n/3$.

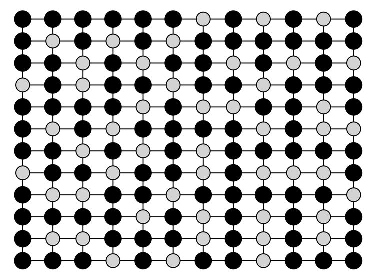




(a) Red-black.



(b) Fixed tree partition.



(c) Variable greedy tree.

Active-Set Identification

- Consider the composite optimization problem,

$$\underset{x \in \mathbb{R}^n}{\text{argmin}} \; f(x) + \sum_{i=1}^n g_i(x_i),$$

where f is strongly-convex, ∇f is Lipschitz-continuous, g_i convex/lower semi-continuous.

→ E.g., non-negative bound constraints, L1-regularized problem

- The subdifferential of each g_i is an interval on the real line and the interior of each ∂g_i at x_i can be written as an open interval,

$$\text{int}\, \partial g_i(x_i) \equiv (l_i, u_i), \qquad (1)$$

where $l_i \in \mathbb{R} \cup \{-\infty\}$ and $u_i \in \mathbb{R} \cup \{\infty\}$.

Definition

The *active-set* for separable g is defined as the set $\mathcal{Z} = \{i : \partial g_i(x_i^*) \text{ is not a singleton}\}$.

- By (1), the set $\mathcal{Z}$ includes indices i where x_i^* is equal to the lower bound on x_i, is equal to the upper bound on x_i, or occurs at a non-smooth value of g_i.

→ For L1-regularization, the active-set is the set of indices such that $x_i^* = 0$ (sparsity pattern).

Definition

The *minimum distance to the nearest boundary of the subdifferential* (1) for all $i \in \mathcal{Z}$,

$$\delta := \min_{i \in \mathcal{Z}} \{\min\{-\nabla_i f(x^*) - l_i, u_i + \nabla_i f(x^*)\}\} \qquad (2)$$

→ For non-negative constraints, we have $\delta = \min_{i \in \mathcal{Z}} \nabla_i f(x^*)$.

→ For L1-regularization, we have $\delta = \lambda - \max_{i \in \mathcal{Z}} |\nabla_i f(x^*)|$.

Theorem: Active-Set Identification

Assume $x^k \to x^*$ and x^* is a non-degenerate solution. Then for any proximal coordinate descent method with a step-size of $1/L$ there exists a finite k such that $x_i^k = x_i^*$ for all $i \in \mathcal{Z}$.

Active-Set Complexity

Definition

The *active-set complexity* is the number of iterations required before an algorithm is guaranteed to have reached the active-set.

- By our assumptions, proximal BCD with cyclic/greedy selection achieves linear rate,

$$\|x^k - x^*\| \leq \left(1 - \frac{1}{\kappa}\right)^k \gamma \leq \exp\left(-\frac{k}{\kappa}\right) \gamma. \qquad (3)$$

→ In our Theorem, we show that active-set identification occurs when $\|x^k - x^*\| \leq \delta/2L$.

Theorem: Active-Set Complexity

For any δ as defined in (2), we have $\|x^{\bar{k}} - x^*\| \leq \delta/2L$ after at most $\kappa \log(2L\gamma/\delta)$ iterations. Further, we identify the active-set after an additional t iterations, where t is the number of iterations required after iteration $\bar{k}$ to select all suboptimal x_i with $i \in \mathcal{Z}$ as part of some b.

→ Bound only depends logarithmically on δ.

- if δ is large, then we can expect to identify the active-set very quickly.

→ Can be modified to use other step-sizes and to analyze proximal gradient methods.