Analyzing and Improving Greedy 2-Coordinate Updates for Equality-Constrained Optimization via Steepest Descent in the 1-Norm

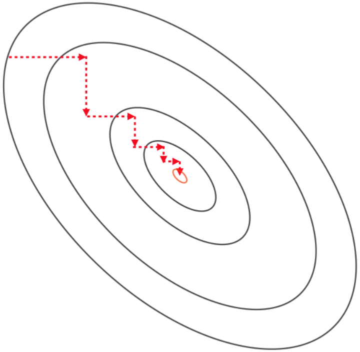
Amrutha Varshini Ramesh, Aaron Mishkin, Mark Schmidt,
Yihan Zhou, Jonathan Wilder Lavington, Jennifer She

University of British Columbia

June 1, 2023

Coordinate Optimization: Theory vs. Practice

- Coordinate optimization updates a small number of variables on each iteration.



- Has convergence rates similar to gradient descent.
- But for some objective functions the iterations have a much lower cost.

Coordinate Optimization: Theory vs. Practice

- A huge literature on the theory of coordinate descent methods.

[PDF] Convergence of a block coordinate descent method for nondifferentiable minimization
P **Tseng** - Journal of optimization theory and applications, 2001 - csie.ntu.edu.tw
We study the convergence properties of a (block) coordinate descent method applied to minimize a nondifferentiable (nonconvex) function f (x1,..., xN) with certain separability and …
☆ Save 99 Cite Cited by 2191 Related articles All 15 versions

Efficiency of **coordinate** descent methods on huge-scale optimization problems
Y **Nesterov** - SIAM Journal on Optimization, 2012 - SIAM
… **coordinate** directional derivatives. The goal of this paper is to provide the random **coordinate** … We show that for functions with cheap **coordinate** derivatives the new methods are always …
☆ Save 99 Cite Cited by 1451 Related articles All 20 versions

Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function
P Richtárik, M Takáč - Mathematical Programming, 2014 - Springer
In this paper we develop a randomized block-coordinate descent method for minimizing the sum of a smooth and a simple nonsmooth block-separable convex function and prove that it …
☆ Save 99 Cite Cited by 809 Related articles All 18 versions

[PDF] Stochastic dual coordinate ascent methods for regularized loss minimization.
S **Shalev-Shwartz**, T **Zhang** - Journal of Machine Learning Research, 2013 - jmlr.org
Stochastic Gradient Descent (SGD) has become popular for solving large scale supervised machine learning optimization problems such as SVM, due to their strong theoretical …
☆ Save 99 Cite Cited by 1144 Related articles All 27 versions

- And most widely-used coordinate optimization code in practice is LIBSVM.
  - Greedy 2-coordinate method for SVM dual (quadratic w/ bounds and sum-to-zero).

LIBSVM: a library for support vector machines
CC Chang, CJ Lin - ACM transactions on intelligent systems and …, 2011 - dl.acm.org
… In this article, we present all implementation details of **LIBSVM**. Issues such as solving SVM … of **LIBSVM**. However, this article does not intend to teach the practical use of **LIBSVM**. For …
☆ Save 99 Cite Cited by 53458 Related articles All 27 versions

- But LIBSVM is not motivated by current theory.
  - Which largely focuses separable constraints, and random selection instead of greedy.

Selected Related Work and Overview of Contribution

- Unconstrained coordinate optimization:
  - Nesterov [2012]: non-asymptotic linear rates for random selection (strongly-convex).
    - As fast as previous rates for greedy selection.
  - Nutini et al. [2015]: faster rates than random with greedy selection.
    - For many problems greedy and random have similar cost.
    - Uses that greedy coordinate optimization is steepest descent in 1-norm.
  - Karimi et al. [2016]: relaxes strong-convexity to Polyak-Łojasiewicz functions.
    - Allows linear rates in many important problems like least squares.

- Bound-constrained coordinate optimization ($l \leq x_i \leq u$):
  - Nesterov [2012]: extends random rates to allow bound constraints.
  - Richtarik and Takac [2014]: allows general non-smooth but convex separable term.
  - Karimreddy et al. [2019]: faster rates than random with greedy selection.
    - Relies on most steps being unconstrained, so taking steepest descent step.

Selected Related Work and Overview of Contribution

- Equality-constrained coordinate optimization ($\sum_i x_i = \gamma$):
  - In this setting we must update at least two coordinates.
  - Tseng and Yun [2009]: asymptotic linear rate with greedy selection.
    - But not faster than radom.
  - Necoara et al. [2011]: non-asymptotic rates for random selection of 2 coordinates.
    - Faster rates shown in Fang et al. [2018].
  - Beck [2014]: sublinear rates for greedy selection (convex and non-convex).

- Our contributions:
  - Equality-constrained coordinate optimization:
    - Show equivalence of greedy to steepest descent in 1-norm.
    - Dimension-independent linear convergence rate (faster than random)
  - Equality-constrained and bound-constrained coordinate optimization:
    - Previous rules cannot guarantee non-trivial progress or have high cost.
    - Steepest descent guarantees fast dimension-independent rate with low cost.

Equality-Constrained 2-Coordinate Update with Greedy Selection

- Consider minimizing a twice-differentiable function with an equality,

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{subject to } \sum_{i=1}^{n} x_i = \gamma.$$

- 2-coordinate method: moves coordinate i_k by δ^k and another j_k by $-\delta^k$.
- The coordinate descent variant chooses δ^k as

$$\delta^k = -\frac{\alpha^k}{2}(\nabla_{i_k} f(x^k) - \nabla_{j_k} f(x^k)),$$

for a step size α^k (but you could alternately find optimal α^k or δ^k).

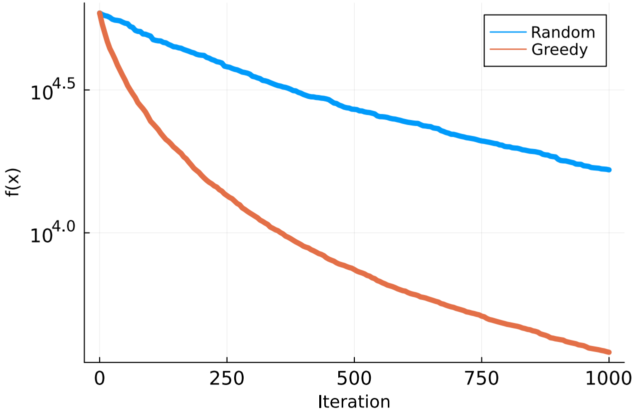
- A greedy rule is to choose coordinates maximizing difference in partial derivatives,

$$i_k \in \operatorname*{argmax}_i \left\{ \nabla_i f(x^k) \right\}, \quad j_k \in \operatorname*{argmin}_j \left\{ \nabla_j f(x^k) \right\},$$

which is sensible because at solution x^* all $\nabla_i f(x^*)$ are equal.

Random Selection vs. Greedy Selection in Practice

- Various random/greedy rules exist, but greedy rules tend to converge faster:



- For the SVM dual problem, random and greedy have the same asymptotic cost.

Connection between Greedy 2-Coordinate Upate and the 1-Norm

- Traditional view of greedy rule is that it is the GS-q rule [Tseng and Yun, 2009],

$$\operatorname*{argmin}_{i,j}\left\{\min_{d_{ij}\in\mathbb{R}^2 | d_i+d_j=0} f(x^k)+\nabla_{ij}f(x^k)^T d_{ij}+\frac{1}{2\alpha^k}\|d_{ij}\|_2^2\right\},$$

  where the coordinates minimize a quadratic approximation.

- Alternate view: we show that greedy rule implements steepest descent in 1-norm,

$$\min_{d\in\mathbb{R}^n | d^T 1=0}\left\{\nabla f(x)^T d+\frac{1}{2(2\alpha)}||d||_1^2\right\}=\min_{i,j}\min_{d_{ij}\in\mathbb{R}^2 | d_i+d_j=0}\left\{\nabla_{ij}f(x)^T d_{ij}+\frac{1}{2\alpha}||d_{ij}||_2^2\right\}$$

  up to a factor of 2 in the step size.

- Proof idea: steepest descent in 1-norm always admits 2-coordinate solution.
  - Can measure progress of 2-coordinate update in terms of a full-coordinate update.

Convergence Rate of Greedy 2-Coordinate Updates under Proximal-PL

Theorem

Let f be a twice-differentiable function whose gradient is 2-coordinate-wise Lipschitz with constant L_2 when restricted to the set where $x^T 1 = \gamma$. If this function satisfies the proximal-PL inequality in the 1-norm for some positive μ_1, then the iterations of the 2-coordinate descent update with $\alpha^k = 1/L_2$ and the greedy rule satisfy:

$$f(x^k) - f(x^*) \leq \left(1 - \frac{2\mu_1}{L_2}\right)^k (f(x^0) - f^*).$$

- Rate for random under same assumptions is dimension-dependent $\left(1 - \frac{\mu_2}{n^2 L_2}\right)^k$.
  - We have $\mu_2/n \leq \mu_1 \leq \mu_2$, so speedup is between n and n^2.
  - Though faster random rates possible for separable f or coordinate-wise Lipschitz.
- Only previous dimension-independent rate for greedy rule is due to Beck [2014].
  - General non-convex problems but sublinear rate.

Equality- and Bound-Constrained 2-Coordinate Updates

- Equality constraints often appear algonside bound constraints as in SVMs,

$$\min_{x\in\mathbb{R}^n} f(x), \quad \text{subject to } \sum_{i=1}^{n} x_i = \gamma, \; l_i \leq x_i \leq u_i.$$

- 2-coordinate descent step in this setting is truncated to stay in the bounds,

$$\delta^k = -\min\left\{\frac{\alpha^k}{2}(\nabla_{i_k} f(x^k) - \nabla_{j_k} f(x^k)), x_{i_k}^k - l_{i_k}, u_{j_k} - x_{j_k}^k\right\},$$

- There are several possible greedy rules in this setting.
  - We will overview the evolution of rules in LIBSVM, then give a new rule.

GS-s Rule: Minimzing Directional Derivative

- The GS-s rule chooses coordinates giving most negative directional derivative,

$$i_k \in \underset{i \,|\, x_i^k > l_i}{\text{argmax}} \left\{ \nabla_i f(x^k) \right\}, \quad j_k \in \underset{j \,|\, x_j^k < u_i}{\text{argmin}} \left\{ \nabla_j f(x^k) \right\},$$

  which is the greedy rule but eliminating steps that immediately violate bounds.
  - First used for SVM by Keerthi et al. [2001], used in LIBSVM up until version 2.7.
- Advantages:
  - Only costs $O(n)$ given gradient.
  - Faster-than-random dimension-independent rate after active-set identified.
  - Fast identification of active set when near solution.
- Disadvantage:
  - Before active set is identified, progress can be arbitrarily slow.
    - Step can be arbitrarily small if you select a coordinate near its boundary.

GS-q Rule: Minimize Quadratic Approximation

- The GS-q rule minimizes a constrained quadratic approximation,

$$\operatorname*{argmin}_{i,j} \left\{ \min_{d_{ij} | d_i + d_j = 0} f(x^k) + \nabla_{ij} f(x^k)^T d_{ij} + \frac{1}{2\alpha^k} \|d_{ij}\|^2 : x^k + d \in [l, u] \right\}.$$

- Advantages:
  - If we only have lower bounds or upper bounds, only costs $O(n)$.
  - Faster-than-random dimension-independent rate after active-set identified.
  - Faster-than-random rate before active-set identified.
- Disadvantages:
  - Non-asymptotic rate is dimension-dependent and slower than asymptotic rate.
  - Slow identification of active set when near solution (if variables near boundary).
  - If you have both lower bounds and upper bounds, costs $O(n^2)$.
- Beginning in version 2.8, LIBSVM uses an approximation of GS-q:
  - First selects a coordinate according to GS-s, then selects one according to GS-q.
  - Only costs $O(n)$ but similar to GS-s progress can be arbitrarily slow.

GS-1 Rule: Steepest Descent in the 1-Norm

- The GS-1 rule performs constrained steepest descent in the 1-norm,

$$d^k \in \underset{l_i \leq x_i + d_i \leq u_i | d^T 1 = 0}{\text{argmin}} \left\{ \nabla f(x^k)^T d + \frac{1}{2\alpha^k} ||d||_1^2 \right\},$$

  previously used by Song et al. [2017] for 1-norm regularized optimization.

- Advantages:
  - Faster-than-random dimension-independent rate (matching asymptotic rate).
  - Fast identification of active set when near solution.
  - We give an algorithm to compute it in $O(n \log n)$.
- Disadvantage:
  - It may require updating more than 2 coordinates in non-asymptotic regime.

Efficient GS-1 Algorithm

- Algorithm for constructing solution to GS-1 rule in $O(n \log n)$:

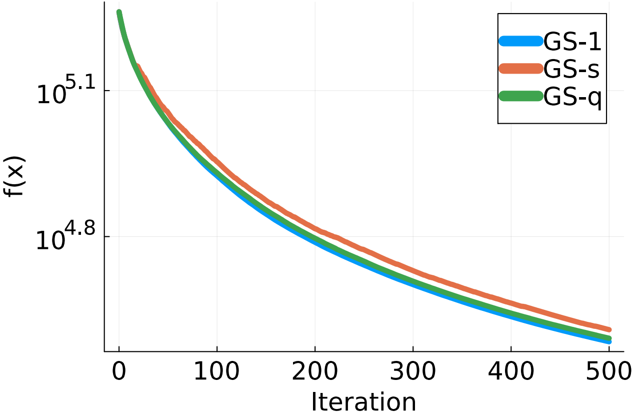
Algorithm 1 The GS-1 algorithm (with variables sorted in descending order according to $\nabla f(x)$).

```
function GS-1(x, ∇f(x), α, l, u)
    x_0 ← 0; x_{n+1} ← 0; i ← 1; j ← n; d ← 0;
    while 1 do
        δ ← (α/4)(∇_i f(x) − ∇_j f(x))
        ω = Σ_{p=0}^{i−1} x_p − l_p; κ = Σ_{q=j+1}^{n+1} u − x_q
        if δ − ω < 0 & δ − κ < 0 then
            if ω < κ then d_i = ω − κ ; break;
            else d_j = ω − κ; break;
            end if
        else if δ − ω < 0 then d_j = ω − κ; break;
        else if δ − κ < 0 then d_i = ω − κ; break;
        end if
        if x_i + ω − δ ≥ l_i & x_j − κ + δ ≤ u_j then
            d_i = ω − δ; d_j = δ − κ; break;
        end if
        if x_i + ω − δ < l_i & x_j − κ + δ > u_j then
            if l_i − (x_i + ω − δ) > x_j − κ + δ − u_j then
                d_i = l − x_i; i ← i + 1
            else
                d_j = u − x_j; j ← j − 1
            end if
        else if x_i + ω − δ < l_i then d_i = l − x_i; i ← i + 1
        else d_j = u − x_j; j ← j − 1
        end if
    end while
    return d
end function
```

- Rough outline of how it satisfies optimality conditions:
  1. If GS-s step does not violate bounds, take it and break.
  2. Move closest variable to boundary and select next largest/smallest $\nabla_i f(x^k)$.
  3. Check whether new variable can overcome 1-norm penalty.
     - If not then "clean up" and break, otherwise go back to 1 with new pair of variables.

Comparing Greedy Rules with Equality and Bounds

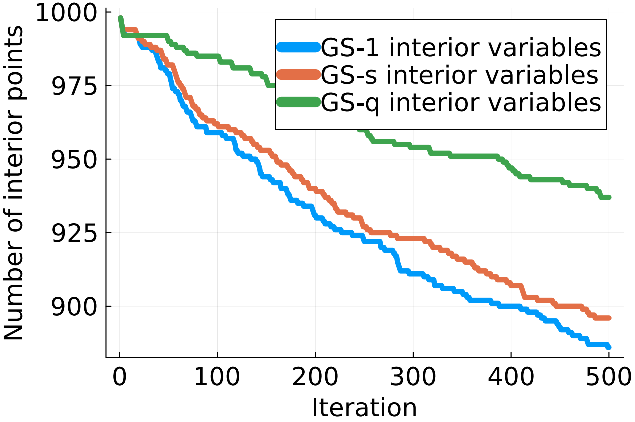
- GS-1 converges slightly faster than GS-q, and both are faster than GS-s.



- Rules were similar in our experiments, but GS-s is much worse on some problems.

Comparing Greedy Rules with Equality and Bounds

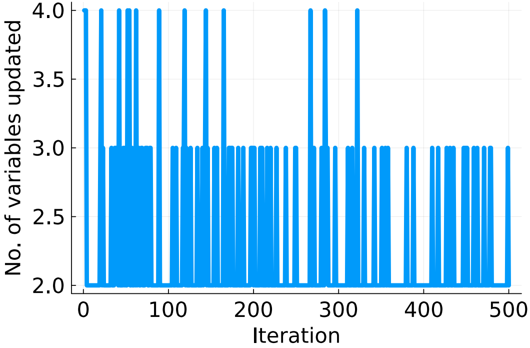
- GS-1 finds active set slightly faster than GS-s, and both are faster than GS-q.

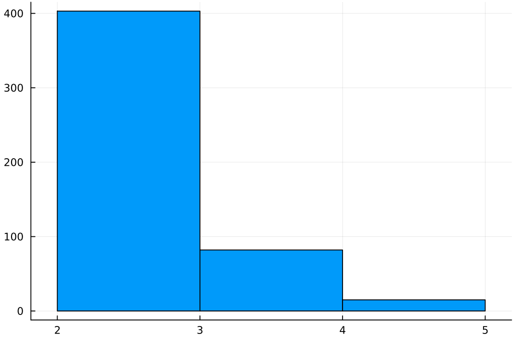


- For SVMs, means identifying support vectors sooner (reducing iteration cost).

Comparing Greedy Rules with Equality and Bounds

- GS-1 updated 2 variables on $> 80\%$ of iterations.





- Rarely updated more than 3, and never more than 4 (out of 1000 variables).

Greedy Updates using Coordinate-Wise Lipschitz Constants

- Instead of blockwise-smoothness, many works use coordinate-wise smoothness,

$$|\nabla_i f(x + \alpha e_i) - \nabla_i f(x)| \leq L_i |\alpha|.$$

- With the summation constraint, the 2-coordinate method with L_i values uses

$$\delta^k = -(\nabla_{i_k} f(x^k) - \nabla_{j_k} f(x^k))/(L_{i_k} + L_{j_k}).$$

- We often analyze coordinate descent methods with L_i-weighted norms, such as

$$\|d\|_L = \sum_{i=1}^{n} \sqrt{L_i}|d_i|,$$

  which can give faster convergence rates.
  - First used by Nesterov [2012] for randomized coordinate descent.
  - First used by Necoara et al. [2011] for 2-coordinate randomized methods.

Different Greedy Rules in Equality-Constrained Case

- The GS-q rule under the L-norm is given by

$$\underset{i,j}{\operatorname{argmax}} \left\{(\nabla_i f(x) - \nabla_j f(x))/\sqrt{L_i + L_j}\right\}.$$

- The GS-1 rule under the L-norm is given by

$$\underset{i,j}{\operatorname{argmax}} \left\{(\nabla_i f(x) - \nabla_j f(x))/(\sqrt{L_i} + \sqrt{L_j})\right\}.$$

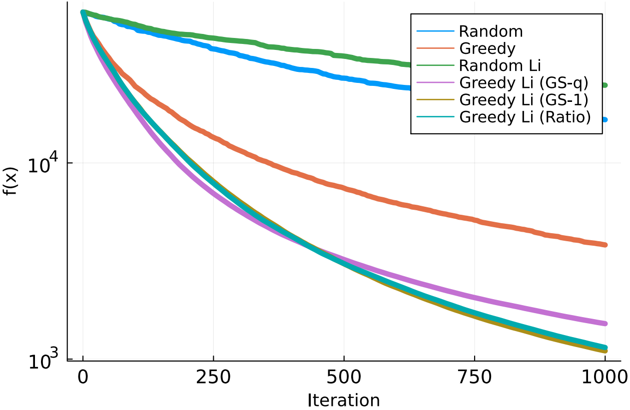
- Thus, the steepest descent equivalence does not hold even without bounds.
  - Both give dimension-independent rates, perform similarly in experiments, cost $O(n^2)$.
- We explored an $O(n)$ approximation:

$$i_k \in \underset{i}{\operatorname{argmax}}(\nabla_i f(x^k) - \mu)/\sqrt{L_i}, \quad j_k \in \underset{j}{\operatorname{argmin}}(\nabla_j f(x^k) - \mu)/\sqrt{L_j},$$

  where μ is the average of the $\nabla_i f(x^k)$ values.
  - Guarantees we choose a coordinate that is above and below mean value.
  - Can also show (slower) dimension-independent rate for this rule.

Experiments: GS-q vs. GS-1 vs. Ratio

- We found that the various L_i rules performed similarly.



- But the new ratio rule is cheaper to compute.

Take-Home Messages

- For equality-constrained optimization:
  - Greedy 2-coordinate rule is steepest descent in the 1-norm.
  - Fast/simple dimension-independent analysis.
    - Faster than random by a factor between n and n^2.

- For equality constrained optimization with bound constraints:
  - GS-s rule does not guarantee non-trivial progress.
  - GS-q rule guarantees dimension-dependent progress but is expensive.
  - GS-1 rule guarantees dimension-independent progress and is cheap.
    - But needs to update more than 2 coordinates on some iterations.

- For equality constraints with known coordinate-wise Lipschitz constants:
  - Greedy rule and steepest descent are no longer equivalent.
  - Both guarantee fast dimension-independent rate, but are costly to implement.
  - Ratio rule is cheap to implement and seems effective in practice.