Is Greedy Coordinate Descent a Terrible Algorithm?

Julie Nutini, Mark Schmidt, Issam Laradji,
Michael Friedlander, Hoyt Koepke

University of British Columbia

International Conference on Machine Learning
Lille, France
July 6th-11th, 2015

NSERC CRSNG
Funded by NSERC Canada Graduate Scholarship

Random vs. Greedy

We consider coordinate descent for large-scale optimization.

Random vs. Greedy

We consider coordinate descent for large-scale optimization.

Recent interest began with Nesterov [2012]:

Random vs. Greedy

We consider coordinate descent for large-scale optimization.

Recent interest began with Nesterov [2012]:

- Global convergence rate for randomized i_k selection.

Random vs. Greedy

We consider coordinate descent for large-scale optimization.

Recent interest began with Nesterov [2012]:

- Global convergence rate for randomized i_k selection.
- Faster than gradient descent if iterations n times cheaper.

Random vs. Greedy

We consider coordinate descent for large-scale optimization.

Recent interest began with Nesterov [2012]:

- Global convergence rate for randomized i_k selection.
- Faster than gradient descent if iterations n times cheaper.

Contrast random with classic Gauss-Southwell (GS) rule:

$$\operatorname*{argmax}_i |\nabla_i f(x)|.$$

Random vs. Greedy

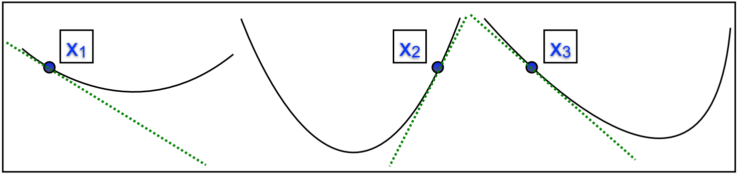
We consider coordinate descent for large-scale optimization.

Recent interest began with Nesterov [2012]:

- Global convergence rate for randomized i_k selection.
- Faster than gradient descent if iterations n times cheaper.

Contrast random with classic Gauss-Southwell (GS) rule:

$$\underset{i}{\operatorname{argmax}} |\nabla_i f(x)|.$$



Random vs. Greedy

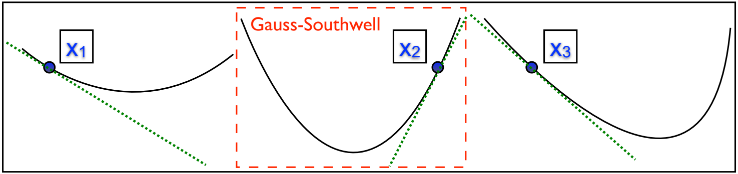
We consider coordinate descent for large-scale optimization.

Recent interest began with Nesterov [2012]:

- Global convergence rate for randomized i_k selection.
- Faster than gradient descent if iterations n times cheaper.

Contrast random with classic Gauss-Southwell (GS) rule:

$$\underset{i}{\text{argmax}}\, |\nabla_i f(x)|.$$



Random vs. Greedy

- GS at least as expensive as random.

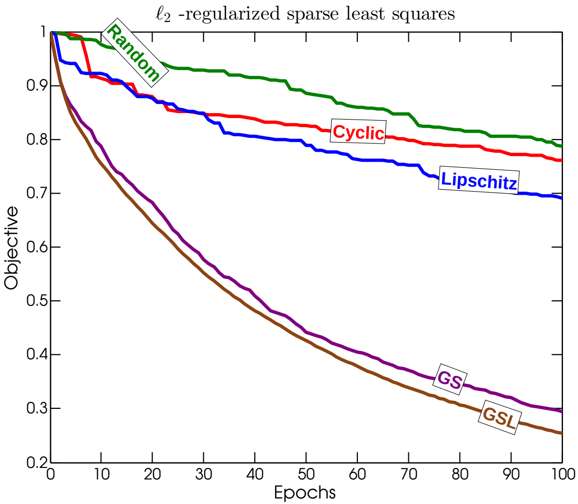
Random vs. Greedy

- GS at least as expensive as random.
- Nesterov showed same rate as random.

Random vs. Greedy

- GS at least as expensive as random.
- Nesterov showed same rate as random.
- But theory disagrees with practice...

Random vs. Greedy



- All rules have similar costs for this problem.

Problems Suitable for Coordinate Descent

Coordinate update n times faster than gradient update for:

Problems Suitable for Coordinate Descent

Coordinate update n times faster than gradient update for:

$$h_1(x) = f(Ax) + \sum_{i=1}^{n} g_i(x_i), \text{ or } h_2(x) = \sum_{i \in V} g_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j)$$

- f and f_{ij} smooth
- A is a matrix
- $\{V, E\}$ is a graph
- g_i general non-degenerate convex functions

Problems Suitable for Coordinate Descent

Coordinate update n times faster than gradient update for:

$$h_1(x) = f(Ax) + \sum_{i=1}^{n} g_i(x_i), \quad \text{or} \quad h_2(x) = \sum_{i \in V} g_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j)$$

- f and f_{ij} smooth
- A is a matrix
- $\{V, E\}$ is a graph
- g_i general non-degenerate convex functions

Examples h_1: least squares, logistic regression, lasso, SVMs.

Problems Suitable for Coordinate Descent

Coordinate update n times faster than gradient update for:

$$h_1(x) = f(Ax) + \sum_{i=1}^{n} g_i(x_i), \text{ or } h_2(x) = \sum_{i \in V} g_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j)$$

- f and f_{ij} smooth
- A is a matrix
- $\{V, E\}$ is a graph
- g_i general non-degenerate convex functions

Examples h_1: least squares, logistic regression, lasso, SVMs.

Examples h_2: quadratics, graph-based label propagation, graphical models.

Problems Suitable for Coordinate Descent

Coordinate update n times faster than gradient update for:

$$h_1(x) = f(Ax) + \sum_{i=1}^{n} g_i(x_i), \quad \text{or} \quad h_2(x) = \sum_{i \in V} g_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j)$$

- f and f_{ij} smooth
- A is a matrix
- $\{V, E\}$ is a graph
- g_i general non-degenerate convex functions

Examples h_1: least squares, logistic regression, lasso, SVMs.

→ Often solvable in $O(cr \log n)$ with c and r non-zeros per column/row.

Examples h_2: quadratics, graph-based label propagation, graphical models.

Problems Suitable for Coordinate Descent

Coordinate update n times faster than gradient update for:

$$h_1(x) = f(Ax) + \sum_{i=1}^{n} g_i(x_i), \quad \text{or} \quad h_2(x) = \sum_{i \in V} g_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j)$$

- f and f_{ij} smooth
- A is a matrix
- $\{V, E\}$ is a graph
- g_i general non-degenerate convex functions

Examples h_1: least squares, logistic regression, lasso, SVMs.

→ Often solvable in $O(cr \log n)$ with c and r non-zeros per column/row.

→ GS rule can be formulated as a maximum inner-product search (MIPS).

Examples h_2: quadratics, graph-based label propagation, graphical models.

Problems Suitable for Coordinate Descent

Coordinate update n times faster than gradient update for:

$$h_1(x) = f(Ax) + \sum_{i=1}^{n} g_i(x_i), \quad \text{or} \quad h_2(x) = \sum_{i \in V} g_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j)$$

- f and f_{ij} smooth
- A is a matrix
- $\{V, E\}$ is a graph
- g_i general non-degenerate convex functions

Examples h_1: least squares, logistic regression, lasso, SVMs.

- → Often solvable in $O(cr \log n)$ with c and r non-zeros per column/row.
- → GS rule can be formulated as a maximum inner-product search (MIPS).

Examples h_2: quadratics, graph-based label propagation, graphical models.

- → GS efficient if maximum degree similar to average degree.

Problems Suitable for Coordinate Descent

Coordinate update n times faster than gradient update for:

$$h_1(x) = f(Ax) + \sum_{i=1}^{n} g_i(x_i), \quad \text{or} \quad h_2(x) = \sum_{i \in V} g_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j)$$

- f and f_{ij} smooth
- A is a matrix
- $\{V, E\}$ is a graph
- g_i general non-degenerate convex functions

Examples h_1: least squares, logistic regression, lasso, SVMs.

→ Often solvable in $O(cr \log n)$ with c and r non-zeros per column/row.

→ GS rule can be formulated as a maximum inner-product search (MIPS).

Examples h_2: quadratics, graph-based label propagation, graphical models.

→ GS efficient if maximum degree similar to average degree.

→ E.g., lattice-structured graphs and complete graphs.

Notation and Assumptions

We focus on the convex optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

Notation and Assumptions

We focus on the convex optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

- ∇f coordinate-wise L-Lipschitz continuous

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

Notation and Assumptions

We focus on the convex optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

- ∇f coordinate-wise L-Lipschitz continuous

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

- f μ-strongly convex, i.e.,

$$x \mapsto f(x) - \frac{\mu}{2}\|x\|^2$$

is convex for some $\mu > 0$.

Notation and Assumptions

We focus on the convex optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

- ∇f coordinate-wise L-Lipschitz continuous

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

- f μ-strongly convex, i.e.,

$$x \mapsto f(x) - \frac{\mu}{2}\|x\|^2$$

  is convex for some $\mu > 0$.

- If f is twice-differentiable, equivalent to

$$\nabla^2_{ii} f(x) \leq L, \qquad \nabla^2 f(x) \succeq \mu \mathbb{I}.$$

Randomized Coordinate Descent

Coordinate descent with constant step-size $\frac{1}{L}$ update:

$$x^{k+1} = x^k - \frac{1}{L}\nabla_{i_k} f(x^k)e_{i_k}, \quad \text{for some } i_k.$$

Randomized Coordinate Descent

Coordinate descent with constant step-size $\frac{1}{L}$ update:

$$x^{k+1} = x^k - \frac{1}{L}\nabla_{i_k} f(x^k) e_{i_k}, \quad \text{for some } i_k.$$

- With i_k chosen uniformly from $\{1, \ldots, n\}$ [Nesterov, 2012],

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{Ln}\right)[f(x^k) - f(x^*)].$$

Randomized Coordinate Descent

Coordinate descent with constant step-size $\frac{1}{L}$ update:

$$x^{k+1} = x^k - \frac{1}{L}\nabla_{i_k} f(x^k)e_{i_k}, \quad \text{for some } i_k.$$

- With i_k chosen uniformly from $\{1, \ldots, n\}$ [Nesterov, 2012],

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{Ln}\right)[f(x^k) - f(x^*)].$$

- Compare to rate of gradient descent,

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L_f}\right)[f(x^k) - f(x^*)].$$

Randomized Coordinate Descent

Coordinate descent with constant step-size $\frac{1}{L}$ update:

$$x^{k+1} = x^k - \frac{1}{L}\nabla_{i_k} f(x^k)e_{i_k}, \quad \text{for some } i_k.$$

- With i_k chosen uniformly from $\{1, \dots, n\}$ [Nesterov, 2012],

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{Ln}\right)[f(x^k) - f(x^*)].$$

- Compare to rate of gradient descent,

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L_f}\right)[f(x^k) - f(x^*)].$$

- Since $Ln \geq L_f \geq L$, coordinate descent is slower *per iteration*, but n coordinate iterations are faster than one gradient iteration.

Classic Analysis: Gauss-Southwell Rule

GS rule chooses coordinate with largest directional derivative,

$$i_k = \mathop{\mathsf{argmax}}_i |\nabla_i f(x^k)|.$$

Classic Analysis: Gauss-Southwell Rule

GS rule chooses coordinate with largest directional derivative,

$$i_k = \underset{i}{\text{argmax}} \, |\nabla_i f(x^k)|.$$

From Lipschitz-continuity assumption this rule satisfies

$$f(x^{k+1}) \leq f(x^k) - \frac{1}{2L} \|\nabla f(x^k)\|_\infty^2.$$

Classic Analysis: Gauss-Southwell Rule

GS rule chooses coordinate with largest directional derivative,

$$i_k = \mathop{\text{argmax}}_i |\nabla_i f(x^k)|.$$

From Lipschitz-continuity assumption this rule satisfies

$$f(x^{k+1}) \leq f(x^k) - \frac{1}{2L}\|\nabla f(x^k)\|_\infty^2.$$

From strong-convexity we have

$$f(x^*) \geq f(x^k) - \frac{1}{2\mu}\|\nabla f(x^k)\|^2.$$

Classic Analysis: Gauss-Southwell Rule

GS rule chooses coordinate with largest directional derivative,

$$i_k = \underset{i}{\text{argmax}}\, |\nabla_i f(x^k)|.$$

From Lipschitz-continuity assumption this rule satisfies

$$f(x^{k+1}) \leq f(x^k) - \frac{1}{2L}\|\nabla f(x^k)\|_\infty^2.$$

From strong-convexity we have

$$f(x^*) \geq f(x^k) - \frac{1}{2\mu}\|\nabla f(x^k)\|^2.$$

Using $\|\nabla f(x^k)\|^2 \leq n\|\nabla f(x^k)\|_\infty^2$ we get

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{Ln}\right)[f(x^k) - f(x^*)].$$

Refined Analysis: Gauss-Southwell Rule

Avoid norm inequality, measure strong-convexity in 1-norm.

Refined Analysis: Gauss-Southwell Rule

Avoid norm inequality, measure strong-convexity in 1-norm.

We now have

$$f(x^*) \geq f(x^k) - \frac{1}{2\mu_1}\|\nabla f(x^k)\|_\infty^2.$$

Refined Analysis: Gauss-Southwell Rule

Avoid norm inequality, measure strong-convexity in 1-norm.

We now have

$$f(x^*) \geq f(x^k) - \frac{1}{2\mu_1}\|\nabla f(x^k)\|_\infty^2.$$

This gives a rate of

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu_1}{L}\right)[f(x^k) - f(x^*)],$$

where

$$\frac{\mu}{n} \leq \mu_1 \leq \mu.$$

Refined Analysis: Gauss-Southwell Rule

Avoid norm inequality, measure strong-convexity in 1-norm.

We now have

$$f(x^*) \geq f(x^k) - \frac{1}{2\mu_1}\|\nabla f(x^k)\|_\infty^2.$$

This gives a rate of

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu_1}{L}\right)[f(x^k) - f(x^*)],$$

where

$$\frac{\mu}{n} \leq \mu_1 \leq \mu.$$

See paper and poster for:

- an explicit formula for μ_1 for separable quadratic;

Refined Analysis: Gauss-Southwell Rule

Avoid norm inequality, measure strong-convexity in 1-norm.

We now have

$$f(x^*) \geq f(x^k) - \frac{1}{2\mu_1}\|\nabla f(x^k)\|_\infty^2.$$

This gives a rate of

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu_1}{L}\right)[f(x^k) - f(x^*)],$$

where

$$\frac{\mu}{n} \leq \mu_1 \leq \mu.$$

See paper and poster for:

- an explicit formula for μ_1 for separable quadratic;
- results showing line-search gives faster rate for sparse problems;

Refined Analysis: Gauss-Southwell Rule

Avoid norm inequality, measure strong-convexity in 1-norm.

We now have

$$f(x^*) \geq f(x^k) - \frac{1}{2\mu_1}\|\nabla f(x^k)\|_\infty^2.$$

This gives a rate of

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu_1}{L}\right)[f(x^k) - f(x^*)],$$

where

$$\frac{\mu}{n} \leq \mu_1 \leq \mu.$$

See paper and poster for:

- an explicit formula for μ_1 for separable quadratic;
- results showing line-search gives faster rate for sparse problems; and
- analysis for approximate Gauss-Southwell rules.

Lipschitz Sampling

Consider the case where we have an L_i for each coordinate

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

and we use a coordinate-dependent step-size,

$$x^{k+1} = x^k - \frac{1}{L_{i_k}}\nabla_{i_k} f(x^k)e_{i_k}.$$

Lipschitz Sampling

Consider the case where we have an L_i for each coordinate

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

and we use a coordinate-dependent step-size,

$$x^{k+1} = x^k - \frac{1}{L_{i_k}} \nabla_{i_k} f(x^k) e_{i_k}.$$

Sampling proportional to L_i yields [Nesterov, 2012]

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{n\bar{L}}\right)[f(x^k) - f(x^*)],$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^n L_i$.

Lipschitz Sampling

Consider the case where we have an L_i for each coordinate

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

and we use a coordinate-dependent step-size,

$$x^{k+1} = x^k - \frac{1}{L_{i_k}} \nabla_{i_k} f(x^k) e_{i_k}.$$

Sampling proportional to L_i yields [Nesterov, 2012]

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{n\bar{L}}\right)[f(x^k) - f(x^*)],$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^{n} L_i$.

- Faster than uniform sampling when L_i are distinct.

Lipschitz Sampling

Consider the case where we have an L_i for each coordinate

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

and we use a coordinate-dependent step-size,

$$x^{k+1} = x^k - \frac{1}{L_{i_k}} \nabla_{i_k} f(x^k) e_{i_k}.$$

Sampling proportional to L_i yields [Nesterov, 2012]

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{n\bar{L}}\right)[f(x^k) - f(x^*)],$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^{n} L_i$.

- Faster than uniform sampling when L_i are distinct.
- Could be faster or slower than GS rule.

Lipschitz Sampling

Consider the case where we have an L_i for each coordinate

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

and we use a coordinate-dependent step-size,

$$x^{k+1} = x^k - \frac{1}{L_{i_k}} \nabla_{i_k} f(x^k) e_{i_k}.$$

Sampling proportional to L_i yields [Nesterov, 2012]

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{n\bar{L}}\right)[f(x^k) - f(x^*)],$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^{n} L_i$.

- Faster than uniform sampling when L_i are distinct.
- Could be faster or slower than GS rule.
- So which should we use?

Lipschitz Sampling

Consider the case where we have an L_i for each coordinate

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

and we use a coordinate-dependent step-size,

$$x^{k+1} = x^k - \frac{1}{L_{i_k}} \nabla_{i_k} f(x^k) e_{i_k}.$$

Sampling proportional to L_i yields [Nesterov, 2012]

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{n\bar{L}}\right)[f(x^k) - f(x^*)],$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^{n} L_i$.

- Faster than uniform sampling when L_i are distinct.
- Could be faster or slower than GS rule.
- So which should we use?
- The answer is neither!

Gauss-Southwell-Lipschitz Rule

We obtain a faster rate by using L_i in the GS rule,

$$i_k = \underset{i}{\text{argmax}} \frac{|\nabla_i f(x^k)|}{\sqrt{L_i}},$$

which we call the Gauss-Southwell-Lipschitz (GSL) rule.

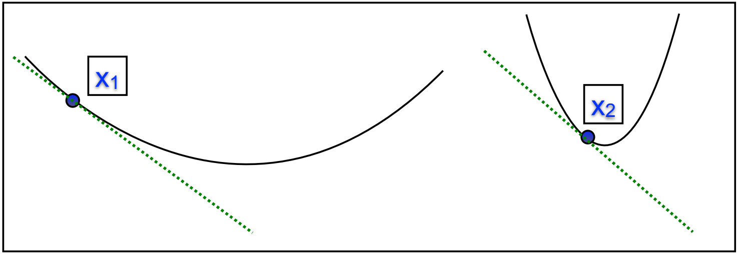
Gauss-Southwell-Lipschitz Rule

We obtain a faster rate by using L_i in the GS rule,

$$i_k = \operatorname*{argmax}_i \frac{|\nabla_i f(x^k)|}{\sqrt{L_i}},$$

which we call the Gauss-Southwell-Lipschitz (GSL) rule.

Intuition: if gradients are similar, more progress if L_i is small.



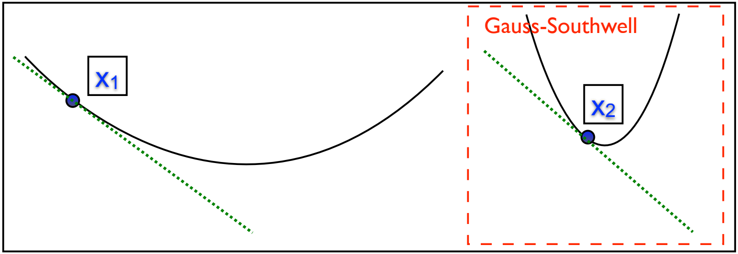
Gauss-Southwell-Lipschitz Rule

We obtain a faster rate by using L_i in the GS rule,

$$i_k = \operatorname*{argmax}_i \frac{|\nabla_i f(x^k)|}{\sqrt{L_i}},$$

which we call the Gauss-Southwell-Lipschitz (GSL) rule.

Intuition: if gradients are similar, more progress if L_i is small.



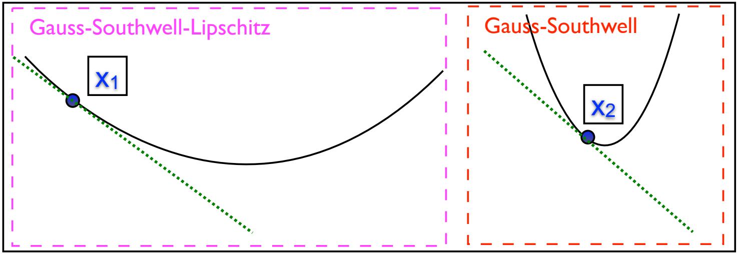
Gauss-Southwell-Lipschitz Rule

We obtain a faster rate by using L_i in the GS rule,

$$i_k = \underset{i}{\text{argmax}} \frac{|\nabla_i f(x^k)|}{\sqrt{L_i}},$$

which we call the Gauss-Southwell-Lipschitz (GSL) rule.

Intuition: if gradients are similar, more progress if L_i is small.



Gauss-Southwell-Lipschitz Rule

The GSL rule obtains a rate of

$$f(x^{k+1}) - f(x^*) \leq (1 - \mu_L)[f(x^k) - f(x^*)],$$

where μ_L satisfies the inequality

$$\max\left\{\frac{\mu}{n\bar{L}}, \frac{\mu_1}{L}\right\} \leq \mu_L \leq \frac{\mu_1}{\min_i\{L_i\}}.$$

Gauss-Southwell-Lipschitz Rule

The GSL rule obtains a rate of

$$f(x^{k+1}) - f(x^*) \leq (1 - \mu_L)[f(x^k) - f(x^*)],$$

where μ_L satisfies the inequality

$$\max\left\{\frac{\mu}{n\bar{L}}, \frac{\mu_1}{L}\right\} \leq \mu_L \leq \frac{\mu_1}{\min_i\{L_i\}}.$$

- GSL is at least as fast as GS and Lipschitz sampling.

Gauss-Southwell-Lipschitz Rule

The GSL rule obtains a rate of

$$f(x^{k+1}) - f(x^*) \leq (1 - \mu_L)[f(x^k) - f(x^*)],$$

where μ_L satisfies the inequality

$$\max\left\{\frac{\mu}{n\bar{L}}, \frac{\mu_1}{L}\right\} \leq \mu_L \leq \frac{\mu_1}{\min_i\{L_i\}}.$$

- GSL is at least as fast as GS and Lipschitz sampling.
- GSL is unimprovable for quadratic functions using $\frac{1}{L_{i_k}}$,

$$f(x^{k+1}) = \underset{i,\alpha}{\text{argmin}}\{f(x^k + \alpha e_i)\}.$$

Gauss-Southwell-Lipschitz Rule

The GSL rule obtains a rate of

$$f(x^{k+1}) - f(x^*) \leq (1 - \mu_L)[f(x^k) - f(x^*)],$$

where μ_L satisfies the inequality

$$\max\left\{\frac{\mu}{n\bar{L}}, \frac{\mu_1}{L}\right\} \leq \mu_L \leq \frac{\mu_1}{\min_i\{L_i\}}.$$

- GSL is at least as fast as GS and Lipschitz sampling.
- GSL is unimprovable for quadratic functions using $\frac{1}{L_{i_k}}$,

$$f(x^{k+1}) = \underset{i,\alpha}{\text{argmin}}\{f(x^k + \alpha e_i)\}.$$

- Gives tighter bound for maximum improvement rule.

Gauss-Southwell-Lipschitz as Nearest Neighbour

Consider a special case of h_1 (no g_i functions),

$$\min_x h_1(x) = f(Ax).$$

Gauss-Southwell-Lipschitz as Nearest Neighbour

Consider a special case of h_1 (no g_i functions),

$$\min_x h_1(x) = f(Ax).$$

The GS rule has the form

$$i_k = \operatorname*{argmax}_i |a_i^T r(x^k)|.$$

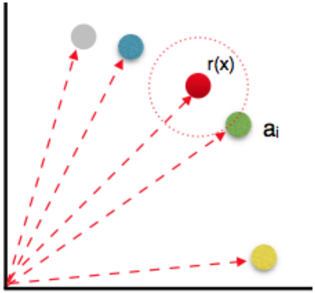
Gauss-Southwell-Lipschitz as Nearest Neighbour

Consider a special case of h_1 (no g_i functions),

$$\min_x h_1(x) = f(Ax).$$

The GS rule has the form

$$i_k = \underset{i}{\text{argmax}}\, |a_i^T r(x^k)|.$$



Gauss-Southwell-Lipschitz as Nearest Neighbour

Dhillon et al. [2011] approximate GS as nearest neighbour,

$$\operatorname*{argmin}_i \|r(x^k) - a_i\| = \operatorname*{argmax}_i \left\{ |\nabla_i f(x^k)| - \frac{1}{2}\|a_i\|^2 \right\}.$$

Gauss-Southwell-Lipschitz as Nearest Neighbour

Dhillon et al. [2011] approximate GS as nearest neighbour,

$$\underset{i}{\text{argmin}}\, \|r(x^k) - a_i\| = \underset{i}{\text{argmax}} \left\{ |\nabla_i f(x^k)| - \frac{1}{2}\|a_i\|^2 \right\}.$$

- Approximation is exact if $\|a_i\| = 1$ for all i.

Gauss-Southwell-Lipschitz as Nearest Neighbour

Dhillon et al. [2011] approximate GS as nearest neighbour,

$$\operatorname*{argmin}_i \|r(x^k) - a_i\| = \operatorname*{argmax}_i \left\{ |\nabla_i f(x^k)| - \frac{1}{2}\|a_i\|^2 \right\}.$$

- Approximation is exact if $\|a_i\| = 1$ for all i.

Usually $L_i = \gamma\|a_i\|^2$, in this case exact GSL is a nearest neighbour problem,

$$\operatorname*{argmin}_i \left\| r(x^k) - \frac{a_i}{\|a_i\|} \right\| = \operatorname*{argmin}_i \left\{ \frac{|\nabla_i f(x^k)|}{\sqrt{L_i}} \right\}.$$

- See paper and poster for numerical results on the nearest neighbour.

Proximal Coordinate Descent

Consider the following problem

$$\min_{x \in \mathbb{R}^n} F(x) \equiv f(x) + \sum_i g_i(x_i),$$

where f is smooth and g_i might be non-smooth.

Proximal Coordinate Descent

Consider the following problem

$$\min_{x \in \mathbb{R}^n} F(x) \equiv f(x) + \sum_i g_i(x_i),$$

where f is smooth and g_i might be non-smooth.

- e.g., ℓ_1-regularization, bound constraints

Proximal Coordinate Descent

Consider the following problem

$$\min_{x \in \mathbb{R}^n} F(x) \equiv f(x) + \sum_i g_i(x_i),$$

where f is smooth and g_i might be non-smooth.

- e.g., ℓ_1-regularization, bound constraints

Apply proximal-gradient style update,

$$x^{k+1} = \textbf{prox}_{\frac{1}{L} g_{i_k}} \left[x^k - \frac{1}{L} \nabla_{i_k} f(x^k) e_{i_k} \right],$$

where

$$\textbf{prox}_{\alpha g}[y] = \operatorname*{argmin}_{x \in \mathbb{R}^n} \frac{1}{2}\|x - y\|^2 + \alpha g(x).$$

Proximal Gauss-Southwell

Several generalizations of GS to this setting:

Proximal Gauss-Southwell

Several generalizations of GS to this setting:

- GS-s: Minimize directional derivative,

$$i_k = \operatorname*{argmax}_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.$$

Proximal Gauss-Southwell

Several generalizations of GS to this setting:

- GS-s: Minimize directional derivative,

$$i_k = \operatorname*{argmax}_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.$$

$\rightarrow$ Commonly-used for ℓ_1-regularization, but $\|x^{k+1} - x^k\|$ could be tiny.

Proximal Gauss-Southwell

Several generalizations of GS to this setting:

- GS-s: Minimize directional derivative,

$$i_k = \underset{i}{\text{argmax}} \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.$$

$\rightarrow$ Commonly-used for ℓ_1-regularization, but $\|x^{k+1} - x^k\|$ could be tiny.

- GS-r: Maximize how far we move,

$$i_k = \underset{i}{\text{argmax}} \left\{ \left| x_i^k - \textbf{prox}_{\frac{1}{L} g_{i_k}} \left[x_i^k - \frac{1}{L} \nabla_{i_k} f(x^k) \right] \right| \right\}.$$

Proximal Gauss-Southwell

Several generalizations of GS to this setting:

- GS-s: Minimize directional derivative,

$$i_k = \operatorname*{argmax}_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.$$

→ Commonly-used for ℓ_1-regularization, but $\|x^{k+1} - x^k\|$ could be tiny.

- GS-r: Maximize how far we move,

$$i_k = \operatorname*{argmax}_i \left\{ \left| x_i^k - \mathbf{prox}_{\frac{1}{L} g_{i_k}} \left[x_i^k - \frac{1}{L} \nabla_{i_k} f(x^k) \right] \right| \right\}.$$

→ Effective for bound constraints, but ignores $g_i(x_i^{k+1}) - g_i(x_i^k)$.

Proximal Gauss-Southwell

Several generalizations of GS to this setting:

- GS-s: Minimize directional derivative,

$$i_k = \operatorname*{argmax}_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.$$

$\rightarrow$ Commonly-used for ℓ_1-regularization, but $\|x^{k+1} - x^k\|$ could be tiny.

- GS-r: Maximize how far we move,

$$i_k = \operatorname*{argmax}_i \left\{ \left| x_i^k - \mathbf{prox}_{\frac{1}{L} g_{i_k}} \left[x_i^k - \frac{1}{L} \nabla_{i_k} f(x^k) \right] \right| \right\}.$$

$\rightarrow$ Effective for bound constraints, but ignores $g_i(x_i^{k+1}) - g_i(x_i^k)$.

- GS-q: Maximize progress under quadratic approximation of f,

$$i_k = \operatorname*{argmin}_i \left\{ \min_d f(x^k) + \nabla_i f(x^k) d + \frac{Ld^2}{2} + g_i(x_i^k + d) - g_i(x_i^k) \right\}.$$

Proximal Gauss-Southwell

Several generalizations of GS to this setting:

- GS-s: Minimize directional derivative,

$$i_k = \underset{i}{\text{argmax}} \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.$$

→ Commonly-used for ℓ_1-regularization, but $\|x^{k+1} - x^k\|$ could be tiny.

- GS-r: Maximize how far we move,

$$i_k = \underset{i}{\text{argmax}} \left\{ \left| x_i^k - \textbf{prox}_{\frac{1}{L} g_{i_k}} \left[x_i^k - \frac{1}{L} \nabla_{i_k} f(x^k) \right] \right| \right\}.$$

→ Effective for bound constraints, but ignores $g_i(x_i^{k+1}) - g_i(x_i^k)$.

- GS-q: Maximize progress under quadratic approximation of f,

$$i_k = \underset{i}{\text{argmin}} \left\{ \min_d f(x^k) + \nabla_i f(x^k) d + \frac{Ld^2}{2} + g_i(x_i^k + d) - g_i(x_i^k) \right\}.$$

→ Least intuitive, but has the best theoretical properties.

Proximal Gauss-Southwell

Several generalizations of GS to this setting:

- GS-s: Minimize directional derivative,

$$i_k = \underset{i}{\text{argmax}} \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.$$

$\rightarrow$ Commonly-used for ℓ_1-regularization, but $\|x^{k+1} - x^k\|$ could be tiny.

- GS-r: Maximize how far we move,

$$i_k = \underset{i}{\text{argmax}} \left\{ \left| x_i^k - \textbf{prox}_{\frac{1}{L} g_{i_k}} \left[x_i^k - \frac{1}{L} \nabla_{i_k} f(x^k) \right] \right| \right\}.$$

$\rightarrow$ Effective for bound constraints, but ignores $g_i(x_i^{k+1}) - g_i(x_i^k)$.

- GS-q: Maximize progress under quadratic approximation of f,

$$i_k = \underset{i}{\text{argmin}} \left\{ \min_d f(x^k) + \nabla_i f(x^k) d + \frac{Ld^2}{2} + g_i(x_i^k + d) - g_i(x_i^k) \right\}.$$

$\rightarrow$ Least intuitive, but has the best theoretical properties.

$\rightarrow$ If you use L_i in the GS-q rule, it is a generalization of GSL rule.

Proximal Gauss-Southwell Convergence Rate

For random selection, Richtárik and Takáč [2014] show

$$\mathbb{E}[F(x^{k+1})] - F(x^k) \leq \left(1 - \frac{\mu}{Ln}\right)[F(x^k) - F(x^*)].$$

Proximal Gauss-Southwell Convergence Rate

For random selection, Richtárik and Takáč [2014] show

$$\mathbb{E}[F(x^{k+1})] - F(x^k) \leq \left(1 - \frac{\mu}{Ln}\right)[F(x^k) - F(x^*)].$$

- the same rate as if non-smooth g_i was not there.

Proximal Gauss-Southwell Convergence Rate

For random selection, Richtárik and Takáč [2014] show

$$\mathbb{E}[F(x^{k+1})] - F(x^k) \leq \left(1 - \frac{\mu}{Ln}\right)[F(x^k) - F(x^*)].$$

- the same rate as if non-smooth g_i was not there.

For the GS-q rule, we show that

$$F(x^{k+1}) - F(x^k) \leq \min\left\{ \left(1 - \frac{\mu}{Ln}\right)[F(x^k) - F(x^*)], \right.$$
$$\left. \left(1 - \frac{\mu_1}{L}\right)[F(x^k) - F(x^*)] + \epsilon_k \right\},$$

where $\epsilon_k \to 0$ measures non-linearity of g_i that are not updated.

Proximal Gauss-Southwell Convergence Rate

For random selection, Richtárik and Takáč [2014] show

$$\mathbb{E}[F(x^{k+1})] - F(x^k) \leq \left(1 - \frac{\mu}{Ln}\right)[F(x^k) - F(x^*)].$$

- the same rate as if non-smooth g_i was not there.

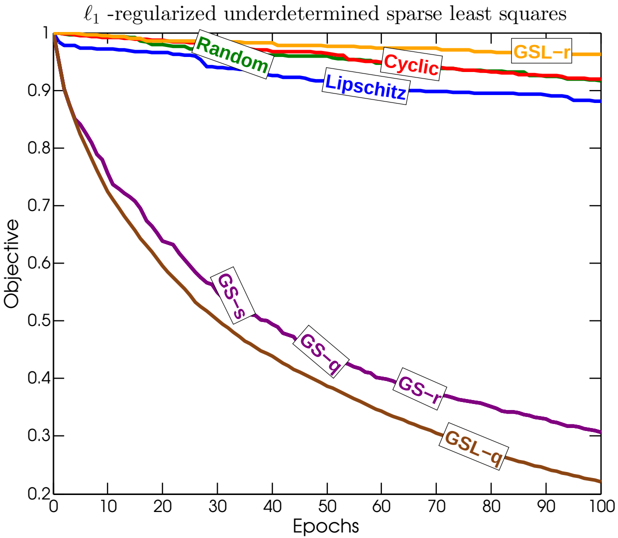
For the GS-q rule, we show that

$$F(x^{k+1}) - F(x^k) \leq \min\left\{ \left(1 - \frac{\mu}{Ln}\right)[F(x^k) - F(x^*)], \right.$$
$$\left. \left(1 - \frac{\mu_1}{L}\right)[F(x^k) - F(x^*)] + \epsilon_k \right\},$$

where $\epsilon_k \to 0$ measures non-linearity of g_i that are not updated.

- But, again theory disagrees with practice...

Comparison of Proximal Gauss-Southwell Rules



Discussion

- GS not always practical.

Discussion

- GS not always practical.
  - But if you can compute GS efficiently, you should use it.

Discussion

- GS not always practical.
  - But if you can compute GS efficiently, you should use it.
- We proposed GSL rule.

Discussion

- GS not always practical.
    - But if you can compute GS efficiently, you should use it.
- We proposed GSL rule.
    - If we know/can approximate L_i, should use GSL.

Discussion

- GS not always practical.
    - But if you can compute GS efficiently, you should use it.
- We proposed GSL rule.
    - If we know/can approximate L_i, should use GSL.
- Analyzed proximal variants of GS rule.

Discussion

- GS not always practical.
    - But if you can compute GS efficiently, you should use it.
- We proposed GSL rule.
    - If we know/can approximate L_i, should use GSL.
- Analyzed proximal variants of GS rule.
    - GSL-q rule least intuitive, has best empirical performance.

Discussion

- GS not always practical.
  - But if you can compute GS efficiently, you should use it.
- We proposed GSL rule.
  - If we know/can approximate L_i, should use GSL.
- Analyzed proximal variants of GS rule.
  - GSL-q rule least intuitive, has best empirical performance.
- See paper and poster for:
  - details on problem types for coordinate descent and GS
  - analysis of μ vs μ_1 for separable quadratic
  - results for exact optimization (chain-structured graph)
  - details on GSL and nearest neighbour analysis
  - convergence rates for approximate GS rules
  - experimental results (e.g., graph-based label propagation)

Discussion

- GS not always practical.
  - But if you can compute GS efficiently, you should use it.
- We proposed GSL rule.
  - If we know/can approximate L_i, should use GSL.
- Analyzed proximal variants of GS rule.
  - GSL-q rule least intuitive, has best empirical performance.
- See paper and poster for:
  - details on problem types for coordinate descent and GS
  - analysis of μ vs μ_1 for separable quadratic
  - results for exact optimization (chain-structured graph)
  - details on GSL and nearest neighbour analysis
  - convergence rates for approximate GS rules
  - experimental results (e.g., graph-based label propagation)
- Current/future work:
  - accelerated/parallel methods [Fercocq & Richtárik, 2013]
  - primal-dual methods [Shalev-Schwartz & Zhang, 2013]
  - without strong-convexity [Luo & Tseng, 1993]

Thank you!