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

Funded by NSERC Canada Graduate Scholarship
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]:

\[ \arg\max_i \left| \nabla f(x) \right|. \]
We consider coordinate descent for large-scale optimization. Recent interest began with Nesterov [2012]:

- Global convergence rate for randomized $i_k$ selection.
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.
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:

$$\arg\max_i |\nabla_i f(x)|.$$
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:

$$\argmax_i |\nabla_i f(x)|.$$
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:

$$\arg\max_i |\nabla_i f(x)|.$$
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...
All rules have similar costs for this problem.
Coordinate update $n$ times faster than gradient update for:

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

- $f$ 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 $O(c r \log n)$ 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.
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
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:

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

- \( f \) 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.

**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:

\[
\begin{align*}
    h_1(x) &= f(Ax) + \sum_{i=1}^{n} g_i(x_i), \quad \text{or} \quad h_2(x) = \sum_{i \in V} g_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j)
\end{align*}
\]

- \( f \) 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.

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

\( \rightarrow \) 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:

\[
\begin{align*}
    h_1(x) &= f(Ax) + \sum_{i=1}^{n} g_i(x_i), \quad \text{or} \quad h_2(x) = \sum_{i \in V} g_i(x_i) + \sum_{(i,j) \in E} f_{ij}(x_i, x_j)
\end{align*}
\]

- \( f \) 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.

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

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

**Examples** \( h_2 \): quadratics, graph-based label propagation, graphical models.

\( \rightarrow \) GS efficient if maximum degree similar to average degree.
Problems Suitable for Coordinate Descent

Coordinate update \( n \) times faster than gradient update for:

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

- \( f \) 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.

- 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.
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

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

- \(\nabla f\) coordinate-wise \(L\)-Lipschitz continuous

\[|\nabla_i f(x + \alpha e_i) - \nabla_i f(x)| \leq L|\alpha|\]
We focus on the convex optimization problem

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

- \( \nabla f \) coordinate-wise \( L \)-Lipschitz continuous

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

- \( f \) \( \mu \)-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

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

- $\nabla f$ coordinate-wise $L$-Lipschitz continuous
  $$|\nabla_i f(x + \alpha e_i) - \nabla_i f(x)| \leq L|\alpha|$$

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

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

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

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

- 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,

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

- 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,

\[ i_k = \arg\max_i |\nabla_i f(x^k)|. \]
Classic Analysis: Gauss-Southwell Rule

GS rule chooses coordinate with largest directional derivative,

\[ i_k = \arg\max_i |\nabla_i f(x^k)|. \]

From Lipschitz-continuity assumption this rule satisfies

\[ f(x^{k+1}) \leq f(x^k) - \frac{1}{2L} \|\nabla f(x^k)\|_\infty^2. \]
GS rule chooses coordinate with largest directional derivative,

\[ i_k = \arg\max_i |\nabla_i f(x^k)|. \]

From Lipschitz-continuity assumption this rule satisfies

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

From strong-convexity we have

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

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

From strong-convexity we have

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

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

\[ f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{Ln}\right)[f(x^k) - f(x^*)]. \]
Avoid norm inequality, measure strong-convexity in 1-norm.

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

This gives a rate of
\[ f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\mu_1}{L}\right) \left[f(x_k) - f(x^*)\right], \]
where \( \mu_n \leq \mu_1 \leq \mu \).

See paper and poster for:
- an explicit formula for \( \mu_1 \) for separable quadratic;
- results showing line-search gives faster rate for sparse problems;
- and analysis for approximate Gauss-Southwell rules.
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

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

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 $\mu_1$ for separable quadratic;
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

\[
\frac{\mu}{\eta} \leq \mu_1 \leq \mu.
\]

See paper and poster for:
- an explicit formula for \(\mu_1\) for separable quadratic;
- results showing line-search gives faster rate for sparse problems;
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 $\mu_1$ for separable quadratic;
- results showing line-search gives faster rate for sparse problems; and
- analysis for approximate Gauss-Southwell rules.
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,

$$x^{k+1} = x^k - \frac{1}{L_i^k} \nabla_i f(x^k) e_{i_k}.$$
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,

\[
x^{k+1} = x^k - \frac{1}{L_i} \nabla_i f(x^k) e_i.
\]

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,

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

Sampling proportional to $L_i$ yields [Nesterov, 2012]

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

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,

$$x^{k+1} = x^k - \frac{1}{\bar{L}_i} \nabla_i f(x^k) e_i.$$

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

$$i_k = \arg \max_i \frac{\left| \nabla_i f(x^k) \right|}{\sqrt{L_i}},$$

which we call the Gauss-Southwell-Lipschitz (GSL) rule.
We obtain a faster rate by using $L_i$ in the GS rule,

$$i_k = \arg\max_i \frac{\left| \nabla_i f(x^k) \right|}{\sqrt{L_i}},$$

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

**Intuition:** if gradients are similar, more progress if $L_i$ is small.
We obtain a faster rate by using $L_i$ in the GS rule,

$$i_k = \arg\max_i \frac{|\nabla_i f(x^k)|}{\sqrt{L_i}},$$

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

**Intuition**: if gradients are similar, more progress if $L_i$ is small.
We obtain a faster rate by using $L_i$ in the GS rule,

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

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

**Intuition**: if gradients are similar, more progress if $L_i$ is small.
The GSL rule obtains a rate of

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

where \( \mu_L \) satisfies the inequality

\[ \max \left\{ \frac{\mu}{n\bar{L}}, \frac{\mu_1}{L} \right\} \leq \mu_L \leq \frac{\mu_1}{\min_i\{L_i\}}. \]
The GSL rule obtains a rate of

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

where \( \mu_L \) 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

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

where \( \mu_L \) 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 \( \frac{1}{L_{i_k}} \),

\[ f(x^{k+1}) = \arg\min_{i, \alpha} \{ f(x^k + \alpha e_i) \}. \]
Gauss-Southwell-Lipschitz Rule

The GSL rule obtains a rate of

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

where \( \mu_L \) 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 \( \frac{1}{L_{ik}} \),

\[ f(x^{k+1}) = \arg\min_{i,\alpha} \{ f(x^k + \alpha e_i) \}. \]

- Gives tighter bound for maximum improvement rule.
Consider a special case of $h_1$ (no $g_i$ functions),

$$\min_x h_1(x) = f(Ax).$$
Consider a special case of $h_1$ (no $g_i$ functions),

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

The GS rule has the form

$$i_k = \arg\max_i |a_i^T r(x^k)|.$$
Consider a special case of $h_1$ (no $g_i$ functions),

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

The GS rule has the form

$$i_k = \arg\max_i |a_i^T r(x^k)|.$$
Dhillon et al. [2011] approximate GS as nearest neighbour,

\[
\arg\min_i \| r(x_k^i) - a_i \| = \arg\min_i \left\{ |\nabla_i f(x_k^i)| - \frac{1}{2} \|a_i\|^2 \right\}.
\]
Dhillon et al. [2011] approximate GS as nearest neighbour,

$$\arg\min_i \| r(x^k) - a_i \| = \arg\min_i \left\{ |\nabla_i f(x^k)| - \frac{1}{2} \|a_i\|^2 \right\}.$$

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

See paper and poster for numerical results on the nearest neighbour.
Dhillon et al. [2011] approximate GS as nearest neighbour,

\[
\arg\min_i \| r(x^k) - a_i \| = \arg\min_i \left\{ |\nabla_i f(x^k)| - \frac{1}{2} \|a_i\|^2 \right\}.
\]

- 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,

\[
\arg\min_i \left\| r(x^k) - \frac{a_i}{\|a_i\|} \right\| = \arg\min_i \left\{ \frac{|\nabla_i f(x^k)|}{\sqrt{L_i}} \right\}.
\]

- See paper and poster for numerical results on the nearest neighbour.
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.
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., $\ell_1$-regularization, bound constraints
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., \( \ell_1 \)-regularization, bound constraints

Apply proximal-gradient style update,

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

where

\[
\text{prox}_{\alpha g}[y] = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2} \| x - y \|^2 + \alpha g(x).
\]
Several generalizations of GS to this setting:

- **GS-**: Minimize directional derivative, $i_k = \arg\max_i \{ \min_s \in \partial g_i |\nabla f(x_k) + s| \}$.
  - Commonly-used for $\ell_1$-regularization, but $\|x_{k+1} - x_k\|$ could be tiny.

- **GS-**: Maximize how far we move, $i_k = \arg\max_i \{|x_{k,i} - \text{prox}_{1Lg_i}[x_{k,i} - 1L\nabla f(x_k)]|\}$.
  - Effective for bound constraints, but ignores $g_i(x_{k+1,i}) - g_i(x_{k,i})$.

- **GS-**: Maximize progress under quadratic approximation of $f$, $i_k = \arg\min_i \{ \min_d df(x_k) + \nabla f(x_k)d + Ld^2 + g_i(x_{k,i} + d) - g_i(x_{k,i}) \}$.
  - Least intuitive, but has the best theoretical properties.
  - If you use $L_i$ in the GS- rule, it is a generalization of GSL rule.
Several generalizations of GS to this setting:

- **GS-$s$**: Minimize directional derivative,
  \[
  i_k = \arg\max_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.
  \]

- **GS-$r$**: Maximize how far we move,
  \[
  i_k = \arg\max_i \left\{ |x_k^i - \text{prox}_1 Lg_{ik}[x_k^i - 1L\nabla_i f(x^k)]| \right\}.
  \]

- **GS-$q$**: Maximize progress under quadratic approximation of $f$,
  \[
  i_k = \arg\min_i \left\{ \min d f(x^k) + \nabla_i f(x^k) d + Ld^2 + g_i(x^k_i + d) - g_i(x^k_i) \right\}.
  \]

→ Least intuitive, but has the best theoretical properties.
→ If you use $L_i$ in the GS-$q$ rule, it is a generalization of GSL rule.
Several generalizations of GS to this setting:

- **GS-$s$:** Minimize directional derivative,
\[
    i_k = \arg\max_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.
\]

→ Commonly-used for $\ell_1$-regularization, but $\|x^{k+1} - x^k\|$ could be tiny.

- **GS-$r$:** Maximize how far we move,
\[
    i_k = \arg\max_i \left\{ \|x^k_i - \text{prox}_{1Lg_i}(x^k) - L\nabla_i f(x^k)\| \right\}.
\]

→ Effective for bound constraints, but ignores $g_i(x^k_i + 1) - g_i(x^k_i)$.

- **GS-$q$:** Maximize progress under quadratic approximation of $f$,
\[
    i_k = \arg\min_i \left\{ \min d f(x^k) + \nabla_i f(x^k) d + Ld^2 + g_i(x^k_i + d) - g_i(x^k_i) \right\}.
\]

→ Least intuitive, but has the best theoretical properties.

→ If you use $L_i$ in the GS-$q$ rule, it is a generalization of GSL rule.
Several generalizations of GS to this setting:

- **GS-\(s\):** Minimize directional derivative,
  \[
  i_k = \arg\max_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\} .
  \]

  → Commonly-used for \(\ell_1\)-regularization, but \(\|x^{k+1} - x^k\|\) could be tiny.

- **GS-\(r\):** Maximize how far we move,
  \[
  i_k = \arg\max_i \left\{ \left\| x_i^k - \text{prox} \left( \frac{1}{L} g_i \right) x_i^k - \frac{1}{L} \nabla_i f(x^k) \right\| \right\} .
  \]
Several generalizations of GS to this setting:

- **GS-**$s$: Minimize directional derivative,

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

→ Commonly-used for $\ell_1$-regularization, but $\|x^{k+1} - x^k\|$ could be tiny.

- **GS-**$r$: Maximize how far we move,

$$i_k = \arg\max_{i} \left\| x_i^k - \text{prox} \frac{1}{L g_i} \left[ x_i^k - \frac{1}{L} \nabla_i f(x^k) \right] \right\}.$$  

→ Effective for bound constraints, but ignores $g_i(x_i^{k+1}) - g_i(x_i^k)$.
Several generalizations of GS to this setting:

- **GS-s**: Minimize directional derivative,
  \[
  i_k = \arg\max_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.
  \]
  → Commonly-used for $\ell_1$-regularization, but $\|x^{k+1} - x^k\|$ could be tiny.

- **GS-r**: Maximize how far we move,
  \[
  i_k = \arg\max_i \left\{ \left| x^k_i - \text{prox}_{\frac{1}{L} g_i} \left[ x^k_i - \frac{1}{L} \nabla_i f(x^k) \right] \right| \right\}.
  \]
  → Effective for bound constraints, but ignores $g_i(x^k_{i+1}) - g_i(x^k_i)$.

- **GS-q**: Maximize progress under quadratic approximation of $f$,
  \[
  i_k = \arg\min_i \left\{ \min_d f(x^k) + \nabla_i f(x^k) d + \frac{L d^2}{2} + g_i(x^k_i + d) - g_i(x^k_i) \right\}.
  \]
Several generalizations of GS to this setting:

- **GS-s**: Minimize directional derivative,
  \[ i_k = \arg \max_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}. \]

  Commonly-used for \( \ell_1 \)-regularization, but \( \|x^{k+1} - x^k\| \) could be tiny.

- **GS-r**: Maximize how far we move,
  \[ i_k = \arg \max_i \left\{ \left| x_i^k - \text{prox}_{\frac{1}{L}g_i} \left[ x_i^k - \frac{1}{L} \nabla_i f(x^k) \right] \right| \right\}. \]

  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 \( f \),
  \[ i_k = \arg \min_{i} \left\{ \min_{d} f(x^k) + \nabla_i f(x^k) d + \frac{Ld^2}{2} + g_i(x_i^k + d) - g_i(x_i^k) \right\}. \]

  Least intuitive, but has the best theoretical properties.
Several generalizations of GS to this setting:

- **GS-\(s\):** Minimize directional derivative,
  \[
  i_k = \argmax_i \left\{ \min_{s \in \partial g_i} |\nabla_i f(x^k) + s| \right\}.
  \]

  → Commonly-used for \(\ell_1\)-regularization, but \(\|x^{k+1} - x^k\|\) could be tiny.

- **GS-\(r\):** Maximize how far we move,
  \[
  i_k = \argmax_i \left\{ \left\| x^k_i - \text{prox} \frac{1}{L} g_{i_k} \left[ x^k_i - \frac{1}{L} \nabla i_k f(x^k) \right] \right\| \right\}.
  \]

  → 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 \(f\),
  \[
  i_k = \argmin_i \left\{ \min_d f(x^k) + \nabla_i f(x^k)d + \frac{Ld^2}{2} + g_i(x_i^k + d) - g_i(x_i^k) \right\}.
  \]

  → Least intuitive, but has the best theoretical properties.

  → If you use \(L_i\) in the GS-\(q\) rule, it is a generalization of GSL rule.
For random selection, Richtárik and Takáč [2014] show

$$\mathbb{E}[F(x^{k+1})] - F(x^k) \leq \left(1 - \frac{\mu}{L}n\right)[F(x^k) - F(x^*)].$$
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 random selection, Richtárik and Takáč [2014] show
\[ E[F(x^{k+1})] - F(x^k) \leq \left( 1 - \frac{\mu}{L\ln(n)} \right) [F(x^k) - F(x^*)]. \]

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

For the GS-$q$ rule, we show that
\[
F(x^{k+1}) - F(x^k) \leq \min \left\{ \left( 1 - \frac{\mu}{L\ln(n)} \right) [F(x^k) - F(x^*)], \right. \\
\left. \left( 1 - \frac{\mu_1}{L} \right) [F(x^k) - F(x^*)] + \epsilon_k \right\},
\]

where $\epsilon_k \to 0$ measures non-linearity of $g_i$ that are not updated.
For random selection, Richtárik and Takáč [2014] show

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

- 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.

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 $\mu$ vs $\mu_1$ 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]
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 $\mu$ vs $\mu_1$ 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 $\mu$ vs $\mu_1$ 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!