# Coordinate descent converges faster with the Gauss-Southwell rule than random selection

**Mark Schmidt**
Department of Computer Science
University of British Columbia

**Michael Friedlander**
Department of Mathematics
University of California, Davis

## Abstract

There has been significant recent work on the theory and application of randomized coordinate-descent algorithms, beginning with the work of Nesterov [*SIAM J. Optim., 22(2), 2012*] who showed that a random-coordinate selection rule achieves the same convergence rate as the Gauss-Southwell selection rule. This suggests that we should never use the Gauss-Southwell rule, which is typically much more expensive than random selection. However, this theoretical result disagrees with the typical empirical behaviours of these algorithms: in applications where both selection rules are computationally cheap, the Gauss-Southwell selection rule tends to perform substantially better than random coordinate selection. We give a simple new analysis of the Gauss-Southwell rule showing that—except in extreme cases cases—it is always faster than choosing random coordinates. Further, under extra assumptions we give a refined Gauss-Southwell rule with an even faster convergence rate.

## 1 Coordinate Descent Methods

There has been substantial recent interest in applying coordinate descent methods to solve large-scale optimization problems. The recent renewal in interest on this topic began with the seminal work of Nesterov [2010, 2012], who gave the first global rate of convergence analysis for coordinate-descent methods for minimizing convex functions. This analysis suggests that choosing at random the coordinate to update gives the same performance as choosing the "best" coordinate to update via the more expensive Gauss-Southwell (GS) rule. (Nesterov further proposed a more clever randomized scheme). This result gives a compelling motivation to use randomized coordinate descent in contexts where the GS rule is too expensive, which has lead to large performance improvements on a variety of problems. However, it also suggests that there is no benefit to using the GS rule in contexts where it is relatively cheap. But in such contexts, the GS rule tends to substantially outperform randomized coordinate selection in practice. This would indicate that either (i) the analysis of GS is not tight, or (ii) there exists a class of functions for which the GS rule is as slow as randomized coordinate descent. After discussing contexts in which it makes sense to use coordinate descent and the GS rule, we answer this theoretical question by giving a tighter analysis of the GS rule (under strong-convexity and standard smoothness assumptions) that yields the same rate as the randomized method for a restricted class of functions, but is otherwise faster (and in some cases substantially faster). Further, in Section 5 we propose a variant of the Gauss-Southwell rule that, similar to Nesterov's more clever randomized sampling scheme, uses knowledge of the Lipschitz constants of the coordinate-wise gradients to obtain a faster rate.

## 2 Problems where we can apply coordinate descent

According to the rates of Nesterov, if we are optimizing $n$ variables then coordinate descent will be faster than gradient descent if we can perform $n$ coordinate updates with a similar cost to one full

gradient iteration. This essentially means that coordinate descent methods are useful for minimizing convex functions that can be expressed in one of the following two forms:

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

where $f$ is smooth and cheap, the $f_{ij}$ are smooth, $\{V, G\}$ is a graph, and $A$ is a matrix. (It is assumed that all functions are convex.)[1] The family $h_1$ includes core machine learning problems like least squares, logistic regression, lasso, and SVMs (when solved in dual form) [Hsieh et al., 2008]; functions in the family $h_2$ include problems like graph-based label propagation algorithms for semi-supervised learning [Bengio et al., 2006] and finding most likely assignments in continuous pairwise graphical models.

In general, the GS rule for these problems is as expensive as a full gradient evaluation, meaning that we should not apply this within coordinate-descent methods. However, additional structure may make the GS rule competitive. For example, Dhillon et al. [2011] show that for the first problem class we can approximate the GS rule by solving a nearest neighbour problem (but their analysis of the GS rule gives the same convergence rate that is obtained by random selection). For the second problem structure, if each node in the graph has only $O(\log n)$ neighbours, we can track the gradients of all the $f_{ij}$ to implement the GS rule with cost a of $O(\log n)$. A similar argument holds for $h_1$ in the case where $A$ has only $O(\log n)$ non-zero values in each column.

## 3 Existing analysis

We are interested in solving the convex optimization problem

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

where $\nabla f$ is coordinate-wise $L$-Lipschitz continuous, meaning that for each $i = 1, \ldots, n$,

$$|\nabla_i f(x + \alpha e_i) - \nabla_i f(x)| \leq L|\alpha|, \quad \forall x \in \mathbb{R}^n \text{ and } \alpha \in \mathbb{R}.$$

For twice-differentiable functions, this is equivalent to the assumption that the diagonal elements of the Hessian are bounded in magnitude by $L$. In contrast, the typical assumption used for gradient methods is that $\nabla f$ is $L^f$-Lipschitz continuous. (Note that $L \leq L^f$, and in the extreme case, $L$ may be up to $n$ times smaller than $L^f$). The coordinate-descent method with constant stepsize is based on the iteration

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

The randomized coordinate selection rule chooses $i_k$ uniformly from the set $\{1, 2, \ldots, n\}$; the GS rule, on the other hand, chooses the coordinate with largest directional derivative,

$$i_k = \arg\max_i \; |\nabla_i f(x^k)|.$$

Under either rule, because $f$ is coordinate-wise Lipschitz continuous, we obtain the following bound on the progress made by each iteration:

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) + \nabla_{i_k} f(x^k)(x^{k+1} - x^k)_{i_k} + \frac{L}{2}(x^{k+1} - x^k)_{i_k}^2 \\ &= f(x^k) - \frac{1}{L}(\nabla_{i_k} f(x^k))^2 + \frac{L}{2}\left[\frac{1}{L}\nabla_{i_k} f(x^k)\right]^2 \\ &= f(x^k) - \frac{1}{2L}[\nabla_{i_k} f(x^k)]^2. \end{aligned} \tag{1}$$

We focus on the case where $f$ is $\mu$-strongly convex meaning that, for some positive $\mu$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2, \quad \forall x, y \in \mathbb{R}^n. \tag{2}$$

We can minimize both sides with respect to $y$ to obtain the bound

$$f(x^*) \geq f(x^k) - \frac{1}{2\mu}\|\nabla f(x^k)\|^2. \tag{3}$$

[1] We could also consider slightly more general cases like functions that are defined on hyper-edges, provided the sparsity in the hyper-graph still allows us to perform $n$ coordinate updates for a similar cost as one gradient evaluation.

### 3.1 Randomized coordinate descent

Conditioning on the $\sigma$-field $\mathcal{F}_{k-1}$ generated by the sequence $\{x^0, x^1, \ldots, x^{k-1}\}$, take expectations of both sides of (1), where $i_k$ is chosen with uniform sampling, to obtain

$$\begin{aligned}\mathbb{E}[f(x^{k+1})] &\leq \mathbb{E}\left[f(x^k) - \frac{1}{2L}\big(\nabla_{i_k} f(x^k)\big)^2\right] \\ &= f(x^k) - \frac{1}{2L}\sum_{i=1}^{n}\frac{1}{n}\big(\nabla_i f(x^k)\big)^2 \\ &= f(x^k) - \frac{1}{2Ln}\|\nabla f(x^k)\|^2.\end{aligned}$$

Using (3) and subtracting $f(x^*)$ from both sides we get

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{Ln}\right)[f(x^k) - f(x^*)]. \tag{4}$$

### 3.2 Gauss-Southwell

We now consider the progress implied by the GS rule. By the definition of $i_k$,

$$(\nabla_{i_k} f(x^k))^2 = \|\nabla f(x^k)\|_\infty^2 \geq (1/n)\|\nabla f(x^k)\|^2. \tag{5}$$

Apply this inequality to (1) to obtain

$$f(x^{k+1}) \leq f(x^k) - \frac{1}{2Ln}\|\nabla f(x^k)\|^2,$$

which together with (3), implies that

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{Ln}\right)[f(x^k) - f(x^*)]. \tag{6}$$

Thus, the (deterministic) GS update yields precisely the same convergence rate as the expected randomized update given by (4).

## 4 Refined Gauss-Southwell analysis

The deficiency of the existing GS analysis is that too much is lost when we use the inequality in (5). To avoid the need to use this inequality, we measure strong-convexity in the 1-norm, i.e.,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\mu_1}{2}\|y - x\|_1^2,$$

which is the analogue of (2). Minimizing both sides with respect to $y$ we get [Nesterov, 2012, §3]

$$\begin{aligned}f(x^*) &\geq f(x) - \sup_y\{\langle -\nabla f(x), y - x\rangle - \frac{\mu_1}{2}\|y - x\|_1^2\} \\ &= f(x) - \left(\frac{\mu_1}{2}\|\cdot\|_1^2\right)^*(-\nabla f(x)) \\ &= f(x) - \frac{1}{2\mu_1}\|\nabla f(x)\|_\infty^2,\end{aligned}$$

which uses that the conjugate $(\frac{\mu_1}{2}\|\cdot\|_1^2)^* = \frac{1}{2\mu_1}\|\cdot\|_\infty^2$. Using this in (1), and the fact that $(\nabla_{i_k} f(x^k))^2 = \|\nabla f(x^k)\|_\infty^2$ for the GS rule, we obtain

$$f(x^{k+1}) - f(x^*) \leq \left(1 - \frac{\mu_1}{L}\right)[f(x^k) - f(x^*)]. \tag{7}$$

It is evident that if $\mu_1 = \mu/n$, then the rates implied by (6) and (7) are identical. However, the rate implied by (7) is faster if $\mu_1 > \mu/n$. It follows from Nesterov [2004, Theorem 2.1.9] that the strong-convexity parameters $\mu$ and $\mu_1$ can be defined by

$$\mu = \inf_{x,y}\frac{\|\nabla f(x) - \nabla f(y)\|}{\|x - y\|} \quad \text{and} \quad \mu_1 = \inf_{x,y}\frac{\|\nabla f(x) - \nabla f(y)\|_\infty}{\|x - y\|_1}.$$

Using the inequalities $\|\cdot\|_\infty \leq \|\cdot\| \leq \|\cdot\|_1$,

$$\mu_1 \equiv \inf_{x,y} \frac{\|\nabla f(x) - \nabla f(y)\|_\infty}{\|x-y\|_1} \leq \inf_{x,y} \frac{\|\nabla f(x) - \nabla f(y)\|}{\|x-y\|} \equiv \mu.$$

Similarly, using the inequalities $\|\cdot\|_1 \leq \sqrt{n}\|\cdot\| \leq n\|\cdot\|_\infty$,

$$\mu \equiv \inf_{x,y} \frac{\|\nabla f(x) - \nabla f(y)\|}{\|x-y\|} \leq \inf_{x,y} \frac{\sqrt{n}\|\nabla f(x) - \nabla f(y)\|_\infty}{\frac{1}{\sqrt{n}}\|x-y\|_1} = n\mu_1.$$

We can summarize these relationships between $\mu$ and $\mu_1$ as

$$\frac{1}{n}\mu \leq \mu_1 \leq \mu.$$

Thus, in extreme cases the GS rule obtains the same rate as uniform selection ($\mu_1 \approx \mu/n$), but on the other extreme it could be faster by a factor of $n$ ($\mu_1 \approx \mu$). That GS only obtains the same rate as random selection in an extreme case seems to explain why the GS rule behaves much better in practice.

We illustrate these two extremes with the simple example of a quadratic function with a diagonal Hessian $\nabla^2 f(x) = \text{diag}(\lambda_1, \ldots, \lambda_n)$. In this case,

$$\mu = \min_i \lambda_i, \quad \text{and} \quad \mu_1 = \frac{\prod_{i=1}^n \lambda_i}{\sum_{k=1}^n \prod_{i \neq k} \lambda_i}.$$

The parameter $\mu_1$ achieves its lower bound when all $\lambda_i$ are equal, $\lambda_1 = \cdots = \lambda_n = \alpha > 0$, in which case

$$\mu = \alpha \quad \text{and} \quad \mu_1 = \alpha/n.$$

Thus, uniform selection does as well as the GS rule if all elements of the gradient change at *exactly* the same rate. This is intuitive, since under this condition there is no apparent advantage in picking the coordinate to update in a clever way. At the other extreme, suppose that $\lambda_1 = \beta$ and $\lambda_2 = \lambda_3 = \cdots = \lambda_n = \alpha$ with $\alpha \geq \beta$. In this case we have

$$\mu = \beta, \quad \text{and} \quad \mu_1 = \frac{\beta\alpha^{n-1}}{\alpha^{n-1} + (n-1)\beta\alpha^{n-2}} = \frac{\beta\alpha}{\alpha + (n-1)\beta}.$$

If we take $\alpha \to \infty$ then we have $\mu_1 \to \beta$ so $\mu_1 \to \mu$. This case is much less intuitive; GS is $n$ times faster than random coordinate selection if one element of the gradient changes much more *slowly* than the others.

## 5 Extensions

If there is a different Lipschitz constant $L_i$ with respect to each coordinate and we choose the coordinate to update proportional to these constants, Nesterov [2010] shows the rate

$$\mathbb{E}[f(x^{k+1})] - f(x^*) \leq \left(1 - \frac{\mu}{\sum_{i=1}^N L_i}\right)[f(x^0) - f(x^*)],$$

which is faster than the rate (4) for uniform sampling. Under our analysis this may or may not be faster than the GS rule. For example, if $\mu_1 = \mu/n$ and any $L_i$ differ then this Lipschitz sampling scheme is faster than our rate for GS. At the other extreme, in our example above with $\alpha$ and $\beta$ the GS and Lipschitz sampling rates are the same when $n = 2$, with a rate of $(1 - \beta/(\alpha+\beta))$. But, the GS rate will be faster for any $\alpha > \beta$ when $n > 2$, since the Lipschitz sampling rate is $(1 - \beta/((n-1)\alpha + \beta))$ which is slower than the GS rate of $(1 - \beta/(\alpha + (n-1)\beta))$. Further, following a similar argument to Section 4, we could use the refined Gauss-Southwell rule $i_k = \arg\max_i \frac{|\nabla_i f(x^k)|}{\sqrt{L_i}}$ to obtain a rate of

$$f(x^{k+1}) - f(x^k) \leq (1 - \mu_1^L)[f(x^0) - f(x^*)].$$

where $\mu_1^L$ is the strong-convexity constant with respect to the norm $\|x\|_1^L = \sum_{i=1}^n \sqrt{L_i}|x_i|$. If all $L_i$ are equal, then $\mu_1^L = \mu_1/L$ and we obtain (7). But otherwise, this refined GS rule gives a faster rate.

Our analysis applies in a straightforward way to block updates by using mixed norms $\|\cdot\|_{p,q}$. We expect that it could also be used with an approximate GS rule [Dhillon et al., 2011], for proximal/accelerated/parallel methods [Fercoq and Richtárik, 2013], for primal-dual rates of dual coordinate ascent Shalev-Schwartz and Zhang [2013], and without strong-convexity under general error bounds [Luo and Tseng, 1993].

## References


Y. Bengio, O. Delalleau, and N. Le Roux. Label propagation and quadratic criterion. *Semi-supervised learning*, pages 193–216, 2006.

I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Nearest neighbor based greedy coordinate descent. *Advances in Neural Information Processing Systems*, 2011.

O. Fercoq and P. Richtárik. Accelerated, parallel and proximal coordinate descent. *arXiv preprint arXiv:1312.5799*, 2013.

C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. *International Conference on Machine Learning*, 2008.

Z.-Q. Luo and P. Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

Y. Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic, Dordrecht, The Netherlands, 2004.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *CORE Discussion Paper*, 2010.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.*, 22(2):341–362, 2012.

S. Shalev-Schwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.