Coordinate Descent

Ryan Tibshirani
Convex Optimization 10-725/36-725

https://www.stat.cmu.edu/~ryantibs/convexopt-S15/lectures/22-coord-desc.pdf

Last time: dual methods and ADMM

Dual methods operate on the dual of a problem that has the form

$$\min_x \ f(x) \ \text{ subject to } \ Ax = b$$

for convex f. The dual (sub)gradient methods chooses an initial $u^{(0)}$, and repeats for $k = 1, 2, 3, \ldots$

$$x^{(k)} \in \operatorname*{argmin}_x \ f(x) + (u^{(k-1)})^T Ax$$
$$u^{(k)} = u^{(k-1)} + t_k(Ax^{(k-1)} - b)$$

where t_k are step sizes, chosen in standard ways

- Pro: decomposability in the first step. Con: poor convergence properties
- Can improve convergence by augmenting the Lagrangian, i.e., add term $\rho/2\|Ax - b\|_2$ to the first step. Perform blockwise minimization $\Rightarrow$ ADMM

Outline

Today:

- Coordinate descent
- Examples
- Implementation tricks
- Coordinate descent—literally
- Screening rules

Coordinate descent

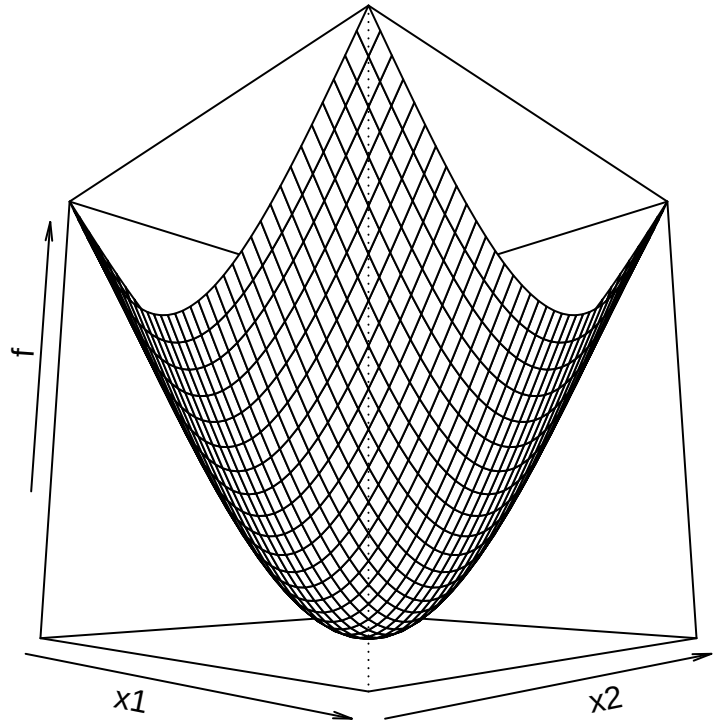
We've seen some pretty sophisticated methods thus far

Our focus today is a very simple technique that can be surprisingly efficient and scalable: coordinate descent, or more appropriately called coordinatewise minimization

Q: Given convex, differentiable $f : \mathbb{R}^n \to \mathbb{R}$, if we are at a point x such that $f(x)$ is minimized along each coordinate axis, then *have we found a global minimizer?*

I.e., does $f(x + \delta e_i) \geq f(x)$ for all $\delta, i \quad \Rightarrow \quad f(x) = \min_z f(z)$?

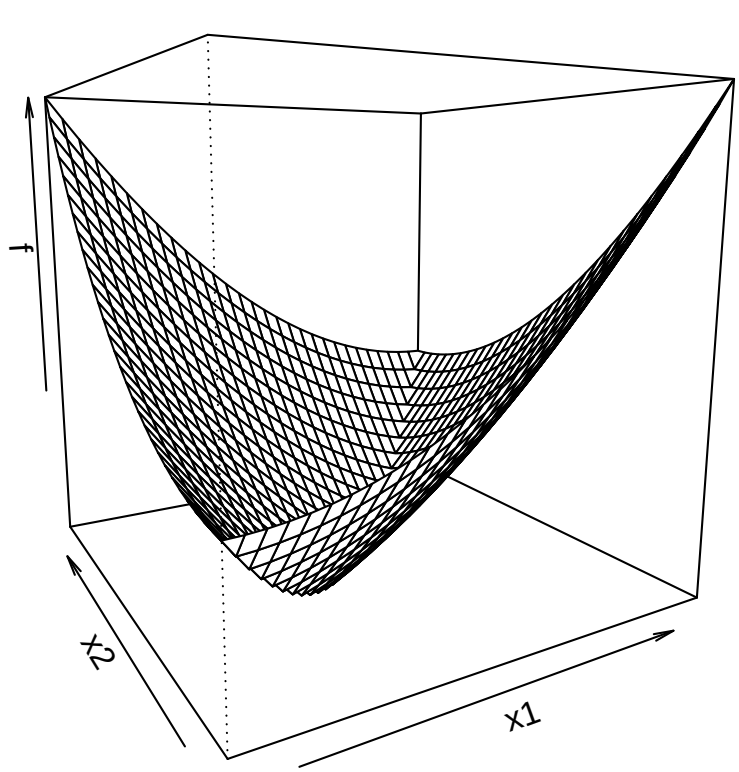
(Here $e_i = (0, \ldots, 1, \ldots 0) \in \mathbb{R}^n$, the ith standard basis vector)

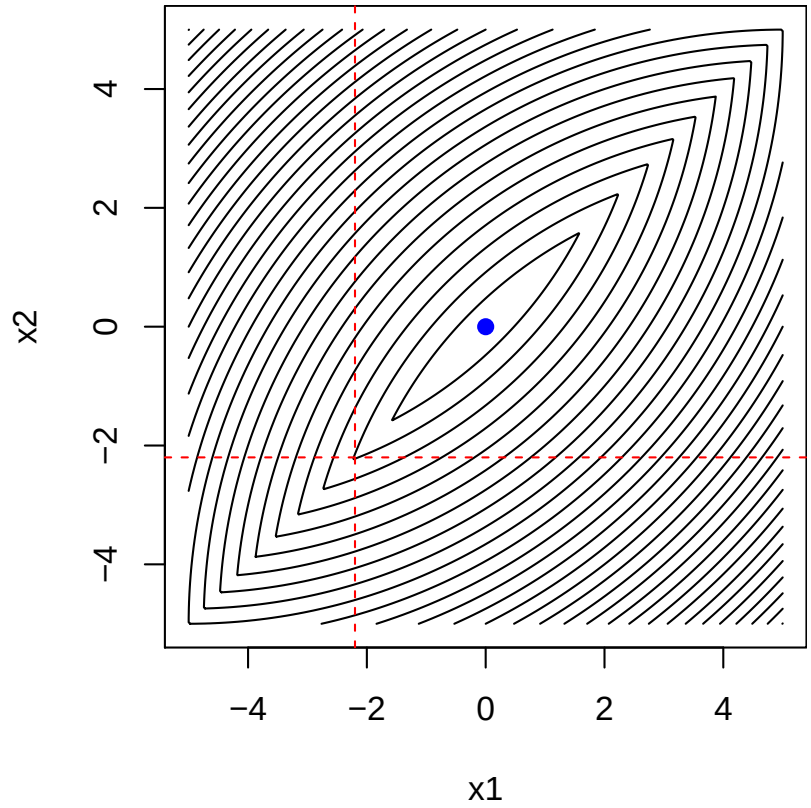


A: Yes! Proof:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \ldots \frac{\partial f}{\partial x_n}(x) \right) = 0$$

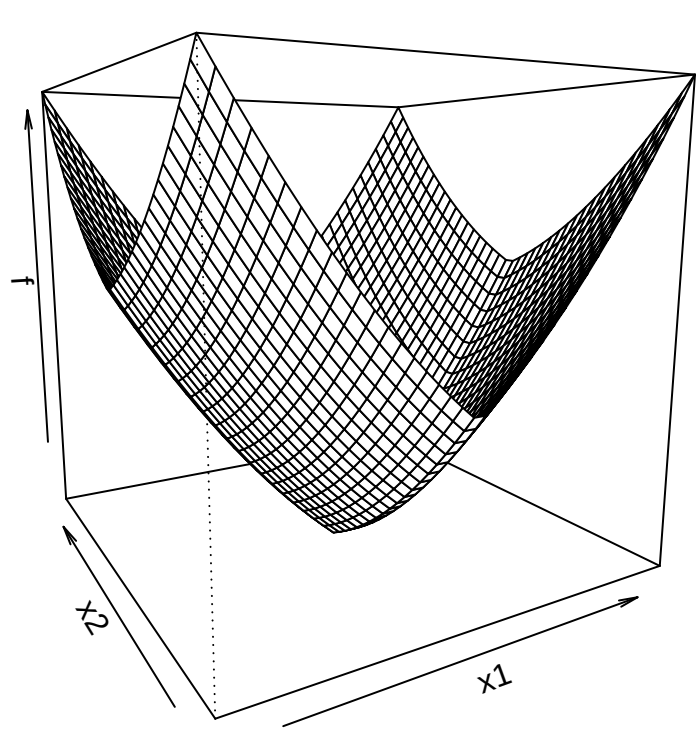
Q: Same question, but for f convex (not differentiable) ... ?

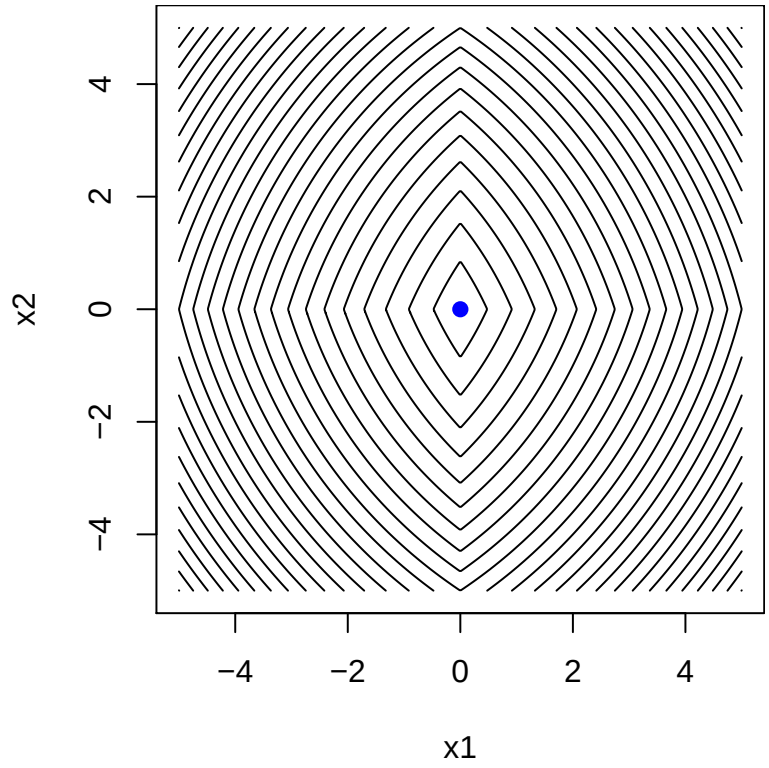




A: No! Look at the above counterexample

Q: Same question again, but now $f(x) = g(x) + \sum_{i=1}^{n} h_i(x_i)$, with g convex, differentiable and each h_i convex ... ? (Nonsmooth part here called separable)





A: Yes! Proof: for any y,

$$f(y) - f(x) \geq \nabla g(x)^T (y - x) + \sum_{i=1}^{n} [h_i(y_i) - h_i(x_i)]$$

$$= \sum_{i=1}^{n} \underbrace{[\nabla_i g(x)(y_i - x_i) + h_i(y_i) - h_i(x_i)]}_{\geq 0} \geq 0$$

Coordinate descent

This suggests that for $f(x) = g(x) + \sum_{i=1}^n h_i(x_i)$ (with g convex, differentiable and each h_i convex) we can use coordinate descent to find a minimizer: start with some initial guess $x^{(0)}$, and repeat

$$
\begin{aligned}
x_1^{(k)} &\in \underset{x_1}{\operatorname{argmin}} \; f\big(x_1, x_2^{(k-1)}, x_3^{(k-1)}, \ldots x_n^{(k-1)}\big) \\
x_2^{(k)} &\in \underset{x_2}{\operatorname{argmin}} \; f\big(x_1^{(k)}, x_2, x_3^{(k-1)}, \ldots x_n^{(k-1)}\big) \\
x_3^{(k)} &\in \underset{x_2}{\operatorname{argmin}} \; f\big(x_1^{(k)}, x_2^{(k)}, x_3, \ldots x_n^{(k-1)}\big) \\
&\ldots \\
x_n^{(k)} &\in \underset{x_2}{\operatorname{argmin}} \; f\big(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \ldots x_n\big)
\end{aligned}
$$

for $k = 1, 2, 3, \ldots$

Note: after we solve for $x_i^{(k)}$, we use its new value from then on!

Tseng (2001) proves that for such f (provided f is continuous on compact set $\{x : f(x) \leq f(x^{(0)})\}$ and f attains its minimum), any limit point of $x^{(k)}$, $k = 1, 2, 3, \ldots$ is a minimizer of f[1]

Notes:

- Order of cycle through coordinates is arbitrary, can use any permutation of $\{1, 2, \ldots n\}$
- Can everywhere replace individual coordinates with blocks of coordinates
- "One-at-a-time" update scheme is critical, and "all-at-once" scheme does not necessarily converge
- For solving linear systems, recall this is exactly the difference between Gauss-Seidel and Jacobi methods

[1] Using real analysis, we know that $x^{(k)}$ has subsequence converging to $x^\star$ (Bolzano-Weierstrass), and $f(x^{(k)})$ converges to $f^\star$ (monotone convergence)

Example: linear regression

Consider linear regression

$$\min_{\beta \in \mathbb{R}^p} \ \frac{1}{2}\|y - X\beta\|_2^2$$

where $y \in \mathbb{R}^n$, and $X \in \mathbb{R}^{n \times p}$ with columns $X_1, \ldots X_p$

Minimizing over β_i, with all β_j, $j \neq i$ fixed:

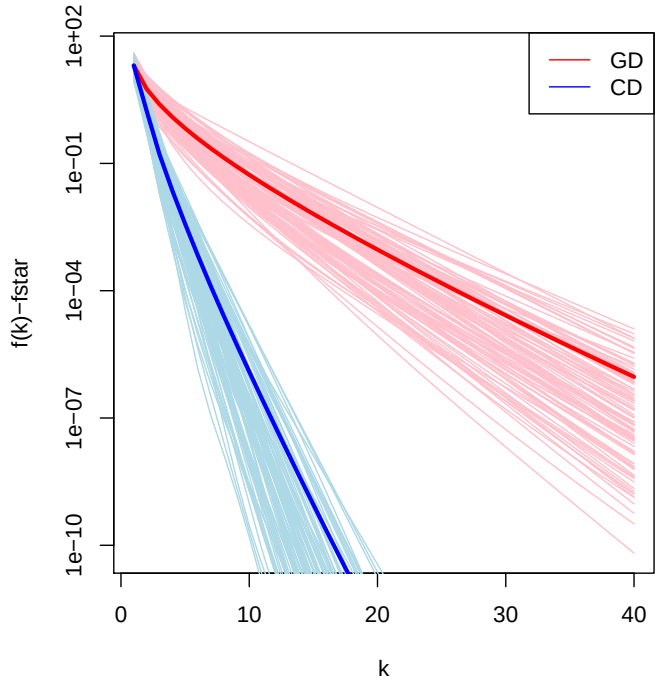
$$0 = \nabla_i f(\beta) = X_i^T(X\beta - y) = X_i^T(X_i\beta_i + X_{-i}\beta_{-i} - y)$$

i.e., we take

$$\beta_i = \frac{X_i^T(y - X_{-i}\beta_{-i})}{X_i^T X_i}$$

Coordinate descent repeats this update for $i = 1, 2, \ldots, p, 1, 2, \ldots$

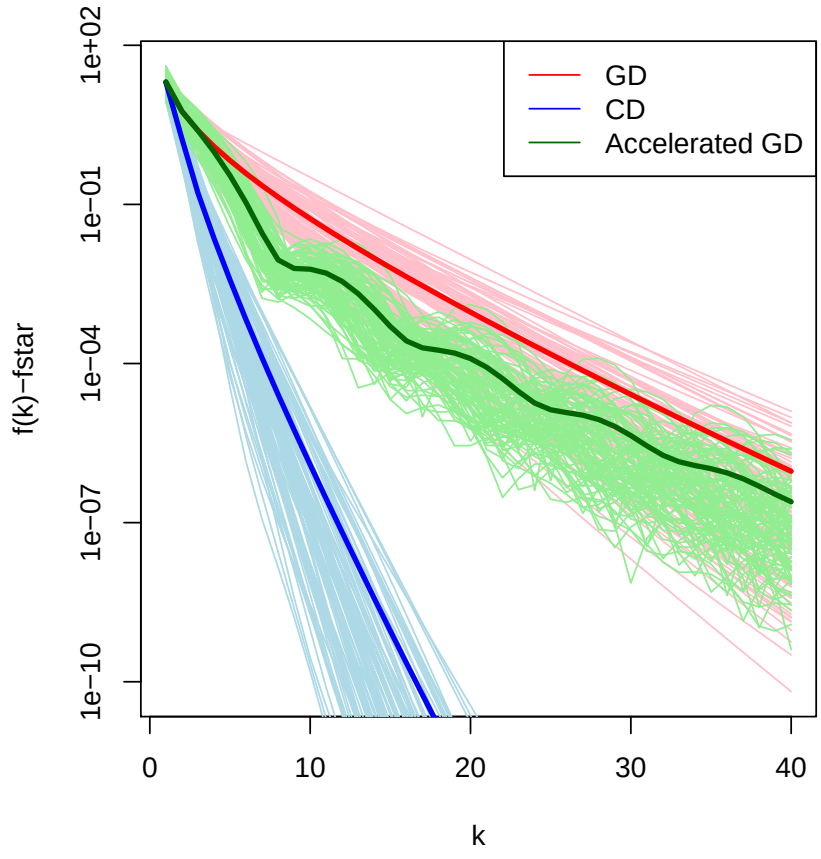
Coordinate descent vs gradient descent for linear regression: 100 instances ($n = 100$, $p = 20$)



Is it fair to compare 1 cycle of coordinate descent to 1 iteration of gradient descent? Yes, if we're clever:

$$\beta_i \leftarrow \frac{X_i^T(y - X_{-i}\beta_{-i})}{X_i^T X_i} = \frac{X_i^T r}{\|X_i\|_2^2} + \beta_i$$

where $r = y - X\beta$. Therefore each coordinate update takes $O(n)$ operations — $O(n)$ to update r, and $O(n)$ to compute $X_i^T r$ — and one cycle requires $O(np)$ operations, just like gradient descent



Same example, but now with accelerated gradient descent for comparison

Is this contradicting the optimality of accelerated gradient descent? I.e., is coordinate descent a first-order method?

No. It uses much more than first-order information

Example: lasso regression

Now consider the lasso problem

$$\min_{\beta \in \mathbb{R}^p} \ \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

Note that the non-smooth part is separable: $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$

Minimizing over β_i, with β_j, $j \neq i$ fixed:

$$0 = X_i^T X_i \beta_i + X_i^T (X_{-i}\beta_{-i} - y) + \lambda s_i$$

where $s_i \in \partial|\beta_i|$. Solution is simply given by soft-thresholding

$$\beta_i = S_{\lambda/\|X_i\|_2^2}\left(\frac{X_i^T(y - X_{-i}\beta_{-i})}{X_i^T X_i}\right)$$

Repeat this for $i = 1, 2, \ldots p, 1, 2, \ldots$

Example: box-constrained regression

Consider box-constrainted linear regression

$$\min_{\beta \in \mathbb{R}^p} \ \frac{1}{2}\|y - X\beta\|_2^2 \ \text{ subject to } \ \|\beta\|_\infty \leq s$$

Note this fits our framework, as $1\{\|\beta\|_\infty \leq s\} = \sum_{i=1}^n 1\{|\beta_i| \leq s\}$

Minimizing over β_i with all β_j, $j \neq i$ fixed: same basic steps give

$$\beta_i = T_s\left(\frac{X_i^T(y - X_{-i}\beta_{-i})}{X_i^T X_i}\right)$$

where T_s is the truncating operator:

$$T_s(u) = \begin{cases} s & \text{if } u > s \\ u & \text{if } -s \leq u \leq s \\ -s & \text{if } u < -s \end{cases}$$

Example: support vector machines

A coordinate descent strategy can be applied to the SVM dual:

$$\min_{\alpha \in \mathbb{R}^n} \ \frac{1}{2}\alpha^T \tilde{X}\tilde{X}^T\alpha - 1^T\alpha \ \text{ subject to } \ 0 \leq \alpha \leq C1, \ \alpha^T y = 0$$

Sequential minimal optimization or SMO (Platt 1998) is basically blockwise coordinate descent in blocks of 2. Instead of cycling, it chooses the next block greedily

Recall the complementary slackness conditions

$$\alpha_i\big(1 - \xi_i - (\tilde{X}\beta)_i - y_i\beta_0\big) = 0, \quad i = 1, \ldots n \tag{1}$$

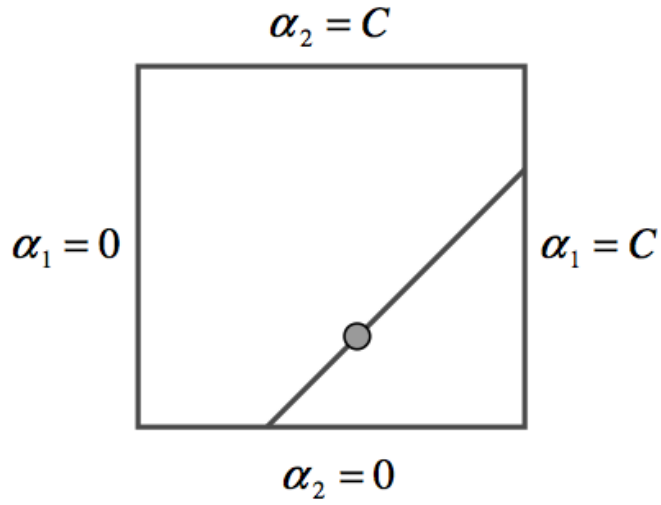
$$(C - \alpha_i)\xi_i = 0, \quad i = 1, \ldots n \tag{2}$$

where β, β_0, ξ are the primal coefficients, intercept, and slacks. Recall that $\beta = \tilde{X}^T\alpha$, β_0 is computed from (1) using any i such that $0 < \alpha_i < C$, and ξ is computed from (1), (2)

SMO repeats the following two steps:

- Choose α_i, α_j that do not satisfy complementary slackness, greedily (using heuristics)
- Minimize over α_i, α_j exactly, keeping all other variables fixed

Using equality constraint, reduces to minimizing univariate quadratic over an interval (From Platt 1998)



Note this does not meet separability assumptions for convergence from Tseng (2001), and a different treatment is required

Many further developments on coordinate descent for SVMs have been made; e.g., a recent one is Hsiesh et al. (2008)

Coordinate descent in statistics and ML

History in statistics:

- Idea appeared in Fu (1998), and again in Daubechies et al. (2004), but was inexplicably ignored
- Three papers around 2007, especially Friedman et al. (2007), really sparked interest in statistics and ML communities

Why is it used?

- Very simple and easy to implement
- Careful implementations can be near state-of-the-art
- Scalable, e.g., don't need to keep full data in memory

Examples: lasso regression, lasso GLMs (under proximal Newton), SVMs, group lasso, graphical lasso (applied to the dual), additive modeling, matrix completion, regression with nonconvex penalties

Pathwise coordinate descent for lasso

Here is the basic outline for pathwise coordinate descent for lasso, from Friedman et al. (2007), Friedman et al. (2009)

Outer loop (pathwise strategy):

- Compute the solution over a sequence $\lambda_1 > \lambda_2 > \ldots > \lambda_r$ of tuning parameter values
- For tuning parameter value λ_k, initialize coordinate descent algorithm at the computed solution for λ_{k+1} (warm start)

Inner loop (active set strategy):

- Perform one coordinate cycle (or small number of cycles), and record active set A of coefficients that are nonzero
- Cycle over coefficients in A until convergence
- Check KKT conditions over all coefficients; if not all satisfied, add offending coefficients to A, go back one step

Even when the solution is only desired at one value of λ, pathwise strategy ($\lambda_1 > \lambda_2 > \ldots > \lambda_r = \lambda$) is typically much more efficient than directly performing coordinate descent at λ

Active set strategy takes advantage of sparsity; e.g., for very large problems, coordinate descent for lasso is much faster than it is for ridge regression

With these strategies in place (and a few more tricks), coordinate descent can be competitve with fastest algorithms for ℓ_1 penalized minimization problems

Freely available via glmnet package in MATLAB or R

What's in a name?

The name coordinate descent is confusing. For a smooth function f, the method that repeats

$$
\begin{aligned}
x_1^{(k)} &= x_1^{(k-1)} - t_{k,1} \cdot \nabla_1 f\big(x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)}, \ldots x_n^{(k-1)}\big) \\
x_2^{(k)} &= x_2^{(k-1)} - t_{k,2} \cdot \nabla_2 f\big(x_1^{(k)}, x_2^{(k-1)}, x_3^{(k-1)}, \ldots x_n^{(k-1)}\big) \\
x_3^{(k)} &= x_3^{(k-1)} - t_{k,3} \cdot \nabla_3 f\big(x_1^{(k)}, x_2^{(k)}, x_3^{(k-1)}, \ldots x_n^{(k-1)}\big) \\
&\ldots \\
x_n^{(k)} &= x_n^{(k-1)} - t_{k,n} \cdot \nabla_n f\big(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \ldots x_n^{(k-1)}\big)
\end{aligned}
$$

for $k = 1, 2, 3, \ldots$ is also (rightfully) called coordinate descent. When $f = g + h$, where g is smooth and h is separable, the proximal version of the above is also called coordinate descent

These versions are often easier to apply that exact coordinatewise minimization, but the latter makes more progress per step

Convergence analyses

Theory for coordinate descent moves quickly. The list given below is incomplete (may not be the latest and greatest). Warning: some references below treat coordinatewise minimization, some do not

- Convergence of coordinatewise minimization for solving linear systems, the Gauss-Seidel method, is a classic topic. E.g., see Golub and van Loan (1996), or Ramdas (2014) for a modern twist that looks at randomized coordinate descent
- Nesterov (2010) considers randomized coordinate descent for smooth functions and shows that it achieves a rate $O(1/\epsilon)$ under a Lipschitz gradient condition, and a rate $O(\log(1/\epsilon))$ under strong convexity
- Richtarik and Takac (2011) extend and simplify these results, considering smooth plus separable functions, where now each coordinate descent update applies a prox operation

- Saha and Tewari (2013) consider minimizing ℓ_1 regularized functions of the form $g(\beta) + \lambda\|\beta\|_1$, for smooth g, and study both cyclic coordinate descent and cyclic coordinatewise min. Under (very strange) conditions on g, they show both methods dominate proximal gradient descent in iteration progress
- Beck and Tetruashvili (2013) study cyclic coordinate descent for smooth functions in general. They show that it achieves a rate $O(1/\epsilon)$ under a Lipschitz gradient condition, and a rate $O(\log(1/\epsilon))$ under strong convexity. They also extend these results to a constrained setting with projections
- The general case of smooth plus separable function is not well-understood with respect to cyclic coordinate descent or cyclic coordinatewise minimization. It is also a question as to whether these two should behave similarly, and whether the aforementioned results are tight ...

Screening rules

In some problems, screening rules can be used in combination with coordinate descent to further wittle down the active set. Screening rules themselves have amassed a sizeable literature recently. Here is an example, the SAFE rule for the lasso[2]:

$$|X_i^T y| < \lambda - \|X_i\|_2 \|y\|_2 \frac{\lambda_{\max} - \lambda}{\lambda_{\max}} \quad \Rightarrow \quad \hat{\beta}_i = 0, \quad \text{all } i = 1, \ldots p$$

where $\lambda_{\max} = \|X^T y\|_\infty$ (the smallest value of λ such that $\hat{\beta} = 0$)

Note: this is not an if and only if statement! But it does give us a way of eliminating features apriori, without solving the lasso

(There have been many advances in screening rules for the lasso, but SAFE is the simplest, and was the first)

[2] El Ghaoui et al. (2010), "Safe feature elimination in sparse learning"

Why is the SAFE rule true? Construction comes from lasso dual:

$$\max_{u \in \mathbb{R}^n} \; g(u) \;\; \text{subject to} \;\; \|X^T u\|_\infty \leq \lambda$$

where $g(u) = \frac{1}{2}\|y\|_2^2 - \frac{1}{2}\|y - u\|_2^2$. Suppose that u_0 is dual feasible (e.g., take $u_0 = y \cdot \lambda/\lambda_{\mathsf{max}}$). Then $\gamma = g(u_0)$ is a lower bound on the dual optimal value, so dual problem is equivalent to

$$\max_{u \in \mathbb{R}^n} \; g(u) \;\; \text{subject to} \;\; \|X^T u\|_\infty \leq \lambda, \; g(u) \geq \gamma$$

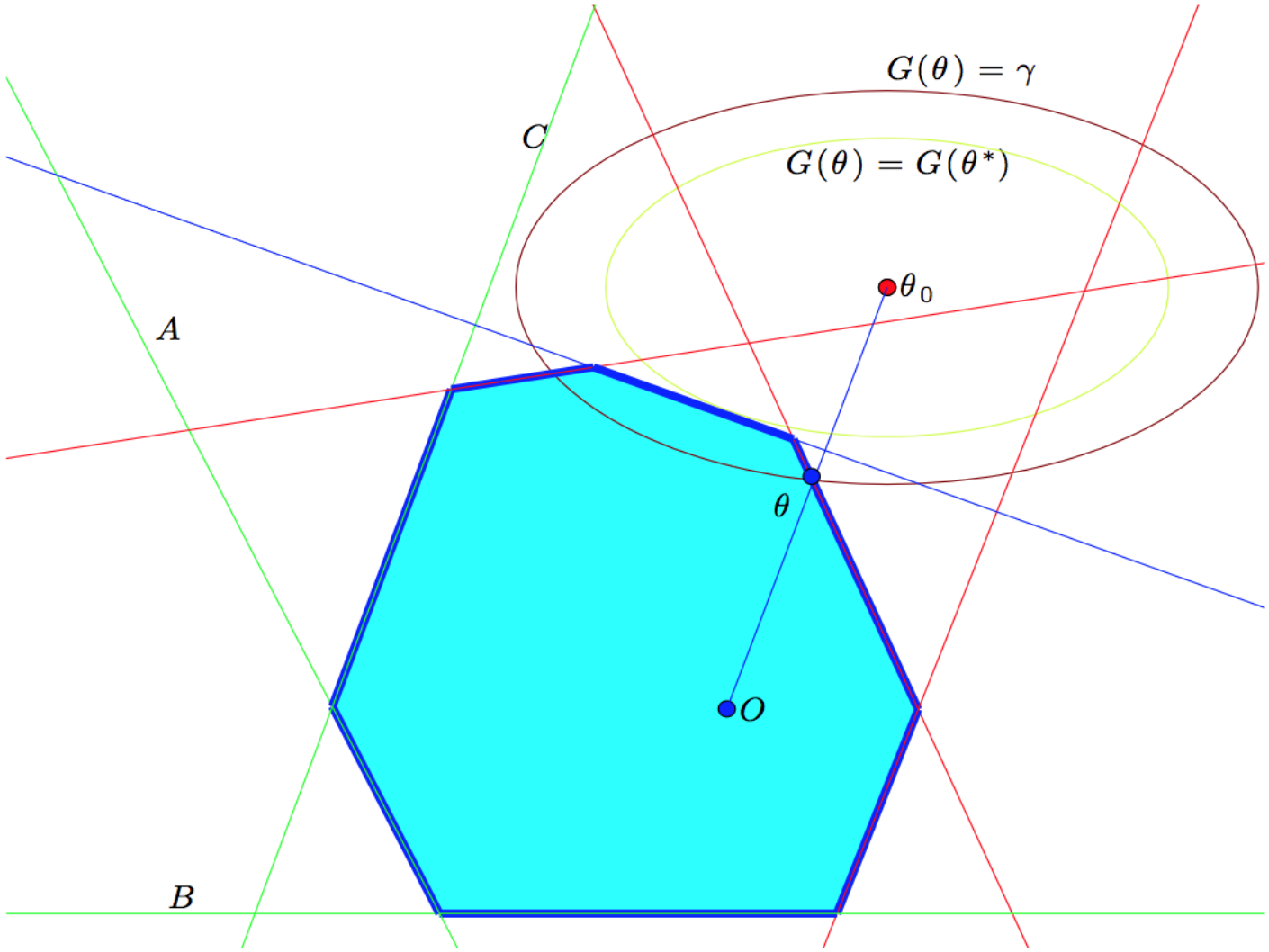
Now consider computing

$$m_i = \max_{u \in \mathbb{R}^n} \; |X_i^T u| \;\; \text{subject to} \;\; g(u) \geq \gamma, \quad \text{for } i = 1, \ldots p$$

Then we would have

$$m_i < \lambda \;\; \Rightarrow \;\; |X_i^T \hat{u}| < \lambda \;\; \Rightarrow \;\; \hat{\beta}_i = 0, \quad i = 1, \ldots p$$

The last implication comes from the KKT conditions



(From El Ghaoui et al. 2010)

Another dual argument shows that

$$\begin{aligned}
&\max_{u \in \mathbb{R}^n} \; X_i^T u \;\; \text{subject to} \;\; g(u) \geq \gamma \\
=&\min_{\mu > 0} \; -\gamma\mu + \frac{1}{\mu}\|\mu y - X_i\|_2^2 \\
=&\|X_i\|_2 \sqrt{\|y\|_2^2 - 2\gamma} - X_i^T y
\end{aligned}$$

where the last equality comes from direct calculation

Thus m_i is given the maximum of the above quantity over $\pm X_i$,

$$m_i = \|X_i\|_2 \sqrt{\|y\|_2^2 - 2\gamma} + |X_i^T y|, \quad i = 1, \ldots p$$

Lastly, subtitute $\gamma = g(y \cdot \lambda / \lambda_{\mathsf{max}})$. Then $m_i < \lambda$ is precisely the safe rule given on previous slide

References

Early coordinate descent references in statistics and ML:

- I. Daubechies and M. Defrise and C. De Mol (2004), "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint"
- J. Friedman and T. Hastie and H. Hoefling and R. Tibshirani (2007), "Pathwise coordinate optimization"
- W. Fu (1998), "Penalized regressions: the bridge versus the lasso"
- T. Wu and K. Lange (2008), "Coordinate descent algorithms for lasso penalized regression"
- A. van der Kooij (2007), "Prediction accuracy and stability of regresssion with optimal scaling transformations"

Applications of coordinate descent:

- O. Banerjee and L. Ghaoui and A. d'Aspremont (2007), "Model selection through sparse maximum likelihood estimation"
- J. Friedman and T. Hastie and R. Tibshirani (2007), "Sparse inverse covariance estimation with the graphical lasso"
- J. Friedman and T. Hastie and R. Tibshirani (2009), "Regularization paths for generalized linear models via coordinate descent"
- C.J. Hsiesh and K.W. Chang and C.J. Lin and S. Keerthi and S. Sundararajan (2008), "A Dual Coordinate Descent Method for Large-scale Linear SVM"
- R. Mazumder and J. Friedman and T. Hastie (2011), "SparseNet: coordinate descent with non-convex penalties"
- J. Platt (1998), "Sequential minimal optimization: a fast algorithm for training support vector machines"

Theory for coordinate descent:

- A. Beck and L. Tetruashvili (2013), "On the convergence of block coordinate descent type methods"
- Y. Nesterov (2010), "Efficiency of coordinate descent methods on huge-scale optimization problems"
- A. Ramdas (2014), "Rows vs columns for linear systems of equations—randomized Kaczmarz or coordinate descent?"
- P. Richtarik and M. Takac (2011), "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function"
- A. Saha and A. Tewari (2013), "On the nonasymptotic convergence of cyclic coordinate descent methods"
- P. Tseng (2001), "Convergence of a block coordinate descent method for nondifferentiable minimization"