10-725/36-725: Convex Optimization **Fall 2015**

Lecture 20: November 7

Lecturer: Ryan Tibshirani *Scribes: Varsha Chinnaobireddy, Joon Sik Kim, Lingyao Zhang*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

20.1 Background of Coordinate Descent

We have studied a lot of sophisticated methods to solve the convex minimization problem, e.g. gradient descent, proximal gradient descent, stochastic gradient descent, Newton's method, Quasi-Newton method, Proximal Newton method, Barrier method, and primal-dual interior point method. These methods are updating the variables from all coordinates at the same time. But these coordinates may not be equally important. It is possible one coordinate influences the criterion value more than other coordinates do. So what if now we can focus on minimizing the criterion according to each coordinate separately? We might be interested in first answering the following questions.

Q: Given convex, differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, if we are at a point x such that $f(x)$ is minimized along each coordinate axis, then have we found a global minimizer? That is, does $f(x + \delta e_i) \geq f(x)$ for all δ, $i \Longrightarrow f(x) = \min_z f(z)$? Note that $e_i = (0, .., 1, ..., 0) \in \mathbb{R}^n$, the ith standard basis vector.

A: Yes! Proof:

$$0 = \nabla f(x) = (\frac{\partial f}{\partial x_1}(x), ..., \frac{\partial f}{\partial x_n}(x)) \tag{20.1}$$

Q: Same question, but now for f convex, and not differentiable?

A: No. Check the counter example in Figure 20.1. If we are now at the intersection of two red lines where the function f is not differentiable, no matter how we move along each axis, we always get larger criterion value. But this is not a global minimum.

Q: Same question again, but now $f(x) = g(x) + h(x) = g(x) + \sum_{i=1}^n h_i(x_i)$, with g convex, differentiable and each h_i convex? (Here the non-smooth part is called separable)

A: Yes! Proof:

Here we want to prove that

$$\forall y \in \mathbb{R}^n, f(y) - f(x) \geq 0 \tag{20.2}$$

We know that

$$f(x + \delta e_i) = g(x + \delta e_i) + \sum_{j \neq i} h_j(x_j) + h_i(x_i + \delta) \tag{20.3}$$

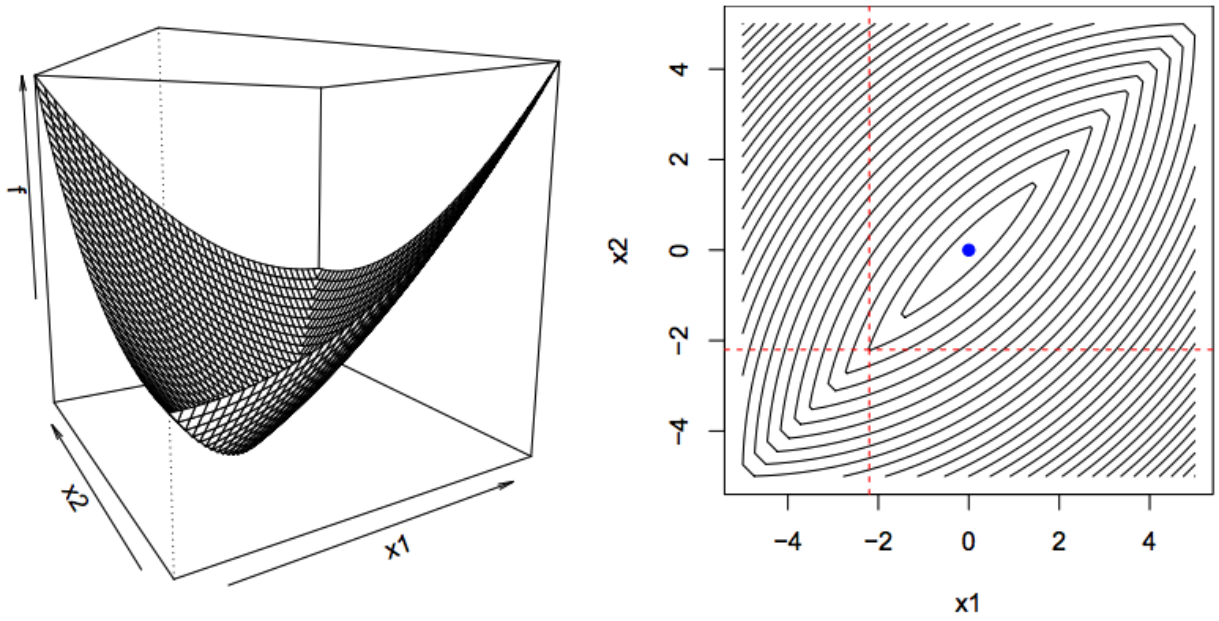

Figure 20.1: A counter example

Since x is optimal along ith axis, according to subgradient optimality, we have

$$\begin{aligned} & 0 \in \nabla_i g(x) + \partial h_i(x_i) \\ \iff & -\nabla_i g(x) \in \partial h_i(x_i) \\ \iff & h_i(y_i) \geq h_i(x_i) - \nabla_i g(x)(y_i - x_i) \\ \iff & \nabla_i g(x)(y_i - x_i) + h_i(y_i) - h_i(x_i) \geq 0 \end{aligned} \tag{20.4}$$

Since f is convex, according to the first-order characterization, we have:

$$\begin{aligned} & f(y) - f(x) \\ & \geq \nabla g(x)^T (y - x) + \sum_{i=1}^{n} [h_i(y_i) - h_i(x_i)] \\ & \geq \sum_{i=1}^{n} [\nabla g_i(x)(y_i - x_i) + h_i(y_i) - h_i(x_i)] \\ & \geq 0 \end{aligned} \tag{20.5}$$

20.2 Coordinate Descent

For the problem

$$\min_x f(x) \tag{20.6}$$

where $f(x) = g(x) + \sum_{i=1}^{n} h_i(x_i)$, with g convex and differentiable and h_i convex, we can use **coordinate descent**:

Let $x^{(0)} \in \mathbb{R}^n$, and for $k = 1, 2, ...$ repeat

$x_i^{(k)} = \texttt{argmin}_{x_i} f(x_1^{(k)}, ..., x_{i-1}^{(k)}, x_i, x_{i+1}^{(k-1)}, ..., x_n^{(k-1)})$, $i = 1, 2, ..., n$

Note that we always use the most recent information possible. Tseng [4] 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 $x^{(k)}$, $k = 1, 2, 3, ...$ is a minimizer of f.

Here are some useful and important notes for coordinate descent:

1. Order of cycle through coordinates is arbitrary, can use any permutation of $\{1, 2, ..., n\}$
2. Can everywhere replace individual coordinates with blocks of coordinates. For example, we can always update a group of coordinates at the same time.
3. "One-at-a-time" update scheme is critical, and "all-at-once" scheme does not necessarily converge.
4. The analogy for solving linear systems: Gauss-Seidel versus Jacobi method.

20.3 Examples of Coordinate Descent

20.3.1 Linear Regression

For the classical linear regression, we consider

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 \tag{20.7}$$

where $y \in \mathbb{R}^n$, and $X \in \mathbb{R}^{n \times p}$. Take the (sub)gradient of the objective with respect to β_i (the ith element of β) where all other $j \neq i$ are fixed and set it to zero to get the update step:

$$X_i^T(X\beta - y) = X_i^T X_i \beta_i + X_i^T(X_{-i}\beta_{-i} - y) = 0 \iff \beta_i \leftarrow \frac{X_i^T(y - X_{-i}\beta_{-i})}{X_i^T X_i} \tag{20.8}$$

where X_{-i} and β_{-i} are original matrix or vector with i-th column or element removed respectively. Repeat this update for $i = 1, 2, ..., p, 1, 2, ...$ This is the same as Guass-Seidl updates.

Remark. The computational cost (in terms of flops) for 1 cycle of coordinate descent is $O(np)$, where $O(n)$ to compute $X_i^T(y - X_{-i}\beta_{-i})$ for each update in a cycle. This is the same as the cost of 1 iteration of gradient descent.

20.3.2 LASSO Regression

For the classical LASSO, we consider

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1 \tag{20.9}$$

where $y \in \mathbb{R}^n$, and $X \in \mathbb{R}^{n \times p}$. Notice that we can use coordinate descent as the regularizer term can be decomposed as the sum of convex functions, namely $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$. Take the (sub)gradient of the objective with respect to β_i where all other $j \neq i$ are fixed and set it to zero to get the update step:

$$X_i^T X_i \beta_i + X_i^T(X_{-i}\beta_{-i} - y) + \lambda s_i = 0 \iff \beta_i \leftarrow S_{\lambda/\|X_i\|_2^2}\left(\frac{X_i^T(y - X_{-i}\beta_{-i})}{X_i^T X_i}\right) \tag{20.10}$$

where $s_i \in \partial|\beta_i|$ and S_λ is a soft-thresholding operator,

$$[S_\lambda(\beta)]_i = \begin{cases} \beta_i - \lambda & \beta_i > \lambda \\ 0 & -\lambda \leq \beta_i \leq \lambda \\ \beta_i + \lambda & \beta_i < -\lambda \end{cases}.$$

Repeat this update for $i = 1, 2, ..., p, 1, 2, ...$

20.3.3 Box-constrained QP

A box-constrained QP has the form:

$$\min_x \frac{1}{2} x^T Q x + b^T x \text{ subject to } l \leq x \leq u \tag{20.11}$$

for $b \in \mathbb{R}^n$, $Q \in \mathbb{S}^n_+$. Notice that we can use coordinate descent as the constraint can be decomposed into element-wise convex constraints: $I(l \leq x \leq u) = \sum_{i=1}^n I(l_i \leq x_i \leq u_i)$, I being the indicator function. Similar steps for taking the (sub)gradient of the objective with respect to x_i with all other elements $j \neq i$ fixed gives the update step:

$$x_i \leftarrow T_{[l_i, u_i]} \left(\frac{b_i - \sum_{j \neq i} Q_{ij} x_j}{Q_{ii}} \right) \tag{20.12}$$

where $T_{[l_i,u_i]}$ is the projection operator on to the interval $[l_i, u_i]$ that clips the value:

$$T_{[l_i,u_i]}(z) = \begin{cases} u_i & z > u_i \\ z & l_i \leq z \leq u_i \\ l_i & z < l_i \end{cases}.$$

Repeat this update for $i = 1, 2, ..., n, 1, 2, ...$

20.3.4 Support Vector Machines

Consider the SVM dual objective:

$$\min_\alpha \frac{1}{2} \alpha^T \tilde{X} \tilde{X}^T \alpha - \mathbf{1}^T \alpha \text{ subject to } 0 \leq C\mathbf{1}, \alpha^T y = 0 \tag{20.13}$$

[3] introduces *Sequential Minimal Optimization* (SMO), a blockwise coordinate descent method that uses greedy heuristics to select the next block of 2 instead of simple cycling. SMO repeats the following updates:

1. Greedily choose a block of i and j such that α_i, α_j violate the complementary slackness condition. That is, select two i's (according to some heuristic) such that

$$\begin{aligned} \alpha_i(1 - \xi_i - (\tilde{X}\beta)_i - y_i\beta_0) &\neq 0 \\ (C - \alpha_i)\xi_i &\neq 0 \end{aligned}$$

   where β, β_0, ξ are primal variables.

2. Minimize the objective over the two chosen variables while keeping others fixed.

For a more recent work on coordinate descent method for SVMs, refer to [2].

20.4 History of Coordinate descent

Until Friedman et. al 2007[1], coordinate descent was considered to be an interesting, toy method. This could be because people were implementing the Jacobian version of it without distinguishing between one at a time versus all at once type of updates.

20.4.1 Why is Coordinate descent used today?

Coordinate descent is very simple and easy to implement. It can achieve state-of-the-art if implemented using some tricks described in the next section. This is especially true for functions in consisting of a quadratic function and a separable component either directly or under proximal Newton. Examples: lasso regression, lasso GLMs (under proximal Newton), SVMs, group lasso, graphical lasso (applied to the dual), etc.

20.5 Implementation tricks - Pathwise Gradient Descent

Pathwise coordinate descent for lasso has the following structure-
Outer Loop(pathwise strategy) : The idea is to go from a sparse to dense solution.

- Compute the solution over a sequence $\lambda_1 > \lambda_2 > ... > \lambda_r$ 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) : This step is efficient since we only work with the active set.

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

Pathwise coordinate descent combined with screening rules make practical coordinate descent very efficient.

20.6 Coordinate gradient descent

For a smooth function f, the iterations

$$x_i^{(k)} = x_i^{(k-1)} - t_{ki}.\nabla_i f(x_1^{(k)}, ..., x_{i-1}^{(k)}, x_i^{(k)}, x_{i+1}^{(k)}, ..., x_n^{(k)}), \quad i = 1...n \tag{20.14}$$

for k = 1, 2, 3, . . . are called coordinate gradient descent, and when f = g + h, with g smooth and $h = \sum_{i=1}^{n} h_i$, the iterations

$$x_i^{(k)} = prox_{h_i,t_{ki}}\left(x_i^{(k-1)} - t_{ki}.\nabla_i g(x_1^{(k)}, ..., x_{i-1}^{(k)}, x_i^{(k)}, x_{i+1}^{(k)}, ..., x_n^{(k)})\right), \quad i = 1...n \tag{20.15}$$

for k = 1, 2, 3, . . . are called coordinate proximal gradient descent. When g is quadratic, (proximal) coordinate gradient descent is the same as coordinate descent under proper step size.
Roughly speaking, theory suggests that the convergence results for coordinate descent are similar to those for proximal gradient descent.

References

[1] Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[2] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, pages 408–415. ACM, 2008.

[3] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[4] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.