Coordinate Descent

Lecturer: Pradeep Ravikumar
Co-instructor: Aarti Singh

Convex Optimization 10-725/36-725

Slides adapted from Tibshirani
Coordinate descent

We’ve seen some pretty sophisticated methods thus far

We now focus on a very simple technique that can be surprisingly efficient, scalable: coordinate descent, or more appropriately called coordinatewise minimization.
Coordinate descent

We’ve seen some pretty sophisticated methods thus far.

We now focus on a very simple technique that can be surprisingly efficient, scalable: coordinate descent, or more appropriately called coordinatewise minimization.

Q: Given convex, differentiable $f : \mathbb{R}^n \rightarrow \mathbb{R}$, 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 $\delta, i \Rightarrow f(x) = \min_z f(z)$?

(Here $e_i = (0, \ldots, 1, \ldots 0) \in \mathbb{R}^n$, the $i$th standard basis vector)
A: Yes! Proof:
\[
f(x) = \sum_{i=1}^{n} f(x_i)
\]

Q: Same question, but now for convex, and not differentiable?
A: Yes! Proof:

\[ \nabla f(x) = \left( \frac{\partial f}{\partial x_1}(x), \ldots, \frac{\partial f}{\partial x_n}(x) \right) = 0 \]
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 now for \( f \) convex, and not differentiable?
A: No! Look at the above counterexample

Q: Same question again, but now $f(x) = g(x) + \sum_{i=1}^{n} h_i(x)$, with $g$ convex, differentiable and each $h_i$ convex? (Here the nonsmooth part is called separable)
A: No! Look at the above counterexample
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? (Here the nonsmooth part is called separable)
A: Yes! Proof: for any $y$, $f(y) - f(x) \leq \text{rg}(x) + \sum_{i=1}^{n} [h_i(y) - h_i(x)] = \sum_{i=1}^{n} [\text{rg}(x)(y_i - x_i) + h_i(y_i) - h_i(x_i)]$.
A: Yes! Proof: for any $y$,

$$f(y) - f(x) \geq \nabla g(x)^T (y - x) + \sum_{i=1}^{n} [h_i(y_i) - h_i(x_i)]$$
A: Yes! Proof: for any $y$,

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

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

$$x_1^{(k)} \in \arg\min_{x_1} f(x_1, x_2^{(k-1)}, x_3^{(k-1)}, \ldots x_n^{(k-1)})$$

$$x_2^{(k)} \in \arg\min_{x_2} f(x_1^{(k)}, x_2, x_3^{(k-1)}, \ldots x_n^{(k-1)})$$

$$x_3^{(k)} \in \arg\min_{x_2} f(x_1^{(k)}, x_2^{(k)}, x_3, \ldots x_n^{(k-1)})$$

$$\ldots$$

$$x_n^{(k)} \in \arg\min_{x_2} f(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \ldots x_n)$$

for $k = 1, 2, 3, \ldots$. Important: after solving 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 $f^1$.
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 $f$.

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
- The analogy for solving linear systems: Gauss-Seidel versus Jacobi method

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

Given $y \in \mathbb{R}^n$, and $X \in \mathbb{R}^{n \times p}$ with columns $X_1, \ldots, X_p$, consider linear regression:

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

Minimizing over $\beta_i$, with all $\beta_j, j \neq i$ 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 with $n = 100, p = 20$
Is it fair to compare 1 cycle of coordinate descent to 1 iteration of gradient descent? Yes, if we’re clever

- Gradient descent: \( \beta \leftarrow \beta + tX^T(y - X\beta) \), costs \( O(np) \) flops
- Coordinate descent, one coordinate update:

\[
\beta_i \leftarrow \frac{X_i^T(y - X_{-i}\beta_{-i})}{X_i^TX_i} = \frac{X_i^Tr}{\|X_i\|^2} + \beta_i
\]

where \( r = y - X\beta \)

- Each coordinate costs \( O(n) \) flops: \( O(n) \) to update \( r \), \( O(n) \) to compute \( X_i^Tr \)
- One cycle of coordinate descent costs \( O(np) \) operations, same as gradient descent
Is this contradicting the optimality of accelerated gradient descent?

Same example, but now with accelerated gradient descent for comparison.
Is this contradicting the optimality of accelerated gradient descent?

No! Coordinate descent uses more than first-order information.
Example: lasso regression

Consider the lasso problem:

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

Note that the nonsmooth part is separable: \(\|\beta\|_1 = \sum_{i=1}^{p} |\beta_i|\)

Minimizing over \(\beta_i\), with \(\beta_j, j \neq i\) 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 \(i = 1, 2, \ldots p, 1, 2, \ldots\)
Proximal gradient vs coordinate descent for lasso regression: 100 instances with $n = 100$, $p = 20$

For same reasons as before:

- All methods use $O(np)$ flops per iteration
- Coordinate descent uses much more information
Example: box-constrained QP

Given $b \in \mathbb{R}^n$, $Q \in \mathbb{S}_+^n$, consider box-constrained quadratic program

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

Fits into our framework, as $I\{l \leq x \leq u\} = \sum_{i=1}^n I\{l_i \leq x_i \leq u_i\}$

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

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

where $T_{[l_i, u_i]}$ is the truncation (projection) operator onto $[l_i, u_i]$: 

$$T_{[l_i, u_i]}(z) = \begin{cases} 
  u_i & \text{if } z > u_i \\
  z & \text{if } l_i \leq z \leq u_i \\
  l_i & \text{if } z < l_i 
\end{cases}$$
Example: support vector machines

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

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

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

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

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

where $\beta, \beta_0, \xi$ are the primal coefficients, intercept, and slacks. Recall that $\beta = \tilde{X}^T \alpha, \ \beta_0$ is computed from (1) using any $i$ such that $0 < \alpha_i < C$, and $\xi$ is computed from (1), (2)
SMO repeats the following two steps:

- Choose $\alpha_i, \alpha_j$ that do not satisfy complementary slackness, greedily (using heuristics)
- Minimize over $\alpha_i, \alpha_j$ 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
What’s in a name?

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

$$
x_1^{(k)} = x_1^{(k-1)} - t_{k,1} \cdot \nabla_1 f(x_1^{(k-1)}, x_2^{(k-1)}, x_3^{(k-1)}, \ldots, x_n^{(k-1)})
$$

$$
x_2^{(k)} = x_2^{(k-1)} - t_{k,2} \cdot \nabla_2 f(x_1^{(k)}, x_2^{(k-1)}, x_3^{(k-1)}, \ldots, x_n^{(k-1)})
$$

$$
x_3^{(k)} = x_3^{(k-1)} - t_{k,3} \cdot \nabla_3 f(x_1^{(k)}, x_2^{(k)}, x_3^{(k-1)}, \ldots, x_n^{(k-1)})
$$

$$
\ldots
$$

$$
x_n^{(k)} = x_n^{(k-1)} - t_{k,n} \cdot \nabla_n f(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \ldots, x_n^{(k-1)})
$$

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

These versions are often easier to apply than 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 $O(1/\varepsilon)$ under a Lipschitz gradient condition, and a rate $O(\log(1/\varepsilon))$ 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 $\ell_1$ 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.

• Nutini et al. (2015) analyze greedy coordinate descent (called Gauss-Southwell rule), and show it achieves a faster rate than randomized coordinate descent for certain problems.

• Wright (2015) provides some unification and a great summary. Also covers parallel versions (even asynchronous ones).

• General theory is still not complete; still unanswered questions (e.g., are descent and minimization strategies the same?)
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:\(^2\):

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

where \(\lambda_{\max} = \|X^T y\|_{\infty}\) (the smallest value of \(\lambda\) 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:

\[
\max_u g(u) \text{ subject to } \|X^T u\|_\infty \leq \lambda
\]

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 \(u_0 = y \cdot \lambda / \lambda_{\text{max}}\)). Then \(\gamma = g(u_0)\) is a lower bound on the dual optimal value, so dual problem is equivalent to

\[
\max_u g(u) \text{ subject to } \|X^T u\|_\infty \leq \lambda, \; g(u) \geq \gamma
\]

Now consider computing

\[
m_i = \max_u |X_i^T u| \text{ subject to } g(u) \geq \gamma, \; \text{for } i = 1, \ldots p
\]

Then we would have

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

The last implication comes from the KKT conditions
Another dual argument shows that

$$\max_u X_i^T u \quad \text{subject to} \quad g(u) \geq \gamma$$

$$= \min_{\mu > 0} -\gamma \mu + \frac{1}{\mu} \| \mu y - X_i \|_2^2$$

$$= \| X_i \|_2 \sqrt{\| y \|_2^2 - 2\gamma} - X_i^T y$$

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, substitute $\gamma = g(y \cdot \lambda / \lambda_{\text{max}})$. Then $m_i < \lambda$ is precisely the safe rule given on previous slide.
Early coordinate descent references in optimization:

Early coordinate descent references in statistics and ML:

- W. Fu (1998), “Penalized regressions: the bridge versus the lasso”
Applications of coordinate descent:

Recent theory for coordinate descent:

- Y. Nesterov (2010), “Efficiency of coordinate descent methods on huge-scale optimization problems”
Screening rules and graphical lasso references:

- R. Mazumder and T. Hastie (2011), “Exact covariance thresholding into connected components for large-scale graphical lasso”