Coordinate Descent

Ryan Tibshirani
Convex Optimization 10-725/36-725
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 $k = 1, 2, 3, \ldots$

$$x^{(k)} \in \arg\min_x f(x) + (u^{(k-1)})^T Ax$$

$$u^{(k)} = u^{(k-1)} + t_k (Ax^{(k-1)} - b)$$

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
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 $f : \mathbb{R}^n \rightarrow \mathbb{R}$, 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:

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

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 \( y \),

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

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

\[
\begin{align*}
    x_1^{(k)} &\in \arg\min_{x_1} f(x_1, x_2^{(k-1)}, x_3^{(k-1)}, \ldots, x_n^{(k-1)}) \\
    x_2^{(k)} &\in \arg\min_{x_2} f(x_1^{(k)}, x_2, x_3^{(k-1)}, \ldots, x_n^{(k-1)}) \\
    x_3^{(k)} &\in \arg\min_{x_2} f(x_1^{(k)}, x_2^{(k)}, x_3, \ldots, x_n^{(k-1)}) \\
    \vdots \\
    x_n^{(k)} &\in \arg\min_{x_2} f(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \ldots, x_n)
\end{align*}
\]

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 \( f \)

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 \( x^* \) (Bolzano-Weierstrass), and \( f(x^{(k)}) \) converges to \( f^* \) (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 $\beta_i$, 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 \((n = 100, p = 20)\)

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

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

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 $$\beta_i$$, 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 $$i = 1, 2, \ldots p, 1, 2, \ldots$$.
Example: box-constrained regression

Consider box-constrained 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 $\beta_i$ with all $\beta_j, j \neq i$ 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:

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

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

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

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

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
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 $\lambda_k$, initialize coordinate descent algorithm at the computed solution for $\lambda_{k+1}$ (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 $\lambda$, pathwise strategy ($\lambda_1 > \lambda_2 > \ldots > \lambda_r = \lambda$) is typically much more efficient than directly performing coordinate descent at $\lambda$

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 competitive with fastest algorithms for $\ell_1$ 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

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

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 $O(1/\epsilon)$ 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 $\ell_1$ 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.

• General rates for cyclic coordinate descent / minimization are not well-understood. Further unknowns: do these two behave similarly? Are the aforementioned results tight?
Graphical lasso

Consider a data matrix $X \in \mathbb{R}^{n \times p}$, whose rows $x_1, \ldots, x_n \in \mathbb{R}^p$ are independent observations from $N(0, \Sigma)$, with unknown covariance matrix $\Sigma$. Want to estimate $\Sigma$.

For $Z \sim N(0, \Sigma)$, normality theory tells us

$$\Sigma_{ij}^{-1} = 0 \iff Z_i, Z_j \text{ conditionally independent given } Z_\ell, \ell \neq i, j$$

If we believe that many components are conditionally independent given others (often reasonable for large $p$) then we want a sparse estimate of $\Sigma^{-1}$. Can get this by solving graphical lasso problem (Banerjee et al. 2007, Friedman et al. 2007):

$$\min_{\Theta \in \mathbb{R}^{p \times p}, \Theta \succ 0} - \log \det \Theta + \text{tr}(S\Theta) + \lambda \|\Theta\|_1$$

Minimizer $\hat{\Theta}$ is an estimate for $\Sigma^{-1}$. (Note here $S = X^T X/n$ is the empirical covariance matrix, and $\|\Theta\|_1 = \sum_{i,j=1}^{p} |\Theta_{ij}|$)
Example from Friedman et al. (2007), cell-signaling network:

Believed network

Graphical lasso estimates

Example from Liu et al. (2010), hub graph simulation:

True graph

Graphical lasso estimate
Graphical lasso KKT conditions (stationarity):

\[-\Theta^{-1} + S + \lambda \Gamma = 0\]

where \(\Gamma_{ij} \in \partial|\Theta_{ij}|\). Let \(W = \Theta^{-1}\); we will solve in terms of \(W\). Note \(W_{ii} = S_{ii} + \lambda\), because \(\Theta_{ii} > 0\) at solution. Now partition:

\[
W = \begin{bmatrix}
W_{11} & w_{12} \\
w_{21} & w_{22}
\end{bmatrix}, \quad \Theta = \begin{bmatrix}
\Theta_{11} & \theta_{12} \\
\theta_{21} & \theta_{22}
\end{bmatrix}, \quad S = \begin{bmatrix}
S_{11} & s_{12} \\
s_{21} & s_{22}
\end{bmatrix}, \quad \Gamma = \begin{bmatrix}
\Gamma_{11} & \gamma_{12} \\
\gamma_{21} & \gamma_{22}
\end{bmatrix}
\]

where \(W_{11} \in \mathbb{R}^{(p-1) \times (p-1)}\), \(w_{12} \in \mathbb{R}^{(p-1) \times 1}\), and \(w_{21} \in \mathbb{R}^{1 \times (p-1)}\), \(w_{22} \in \mathbb{R}\); same with others.

Coordinate descent strategy: solve for \(w_{12}\), the last column of \(W\) (note \(w_{22}\) is known), with all other columns fixed; then solve for second-to-last column, etc., and cycle around until convergence. (Solve for \(\Theta\) along the way, so we don’t have to invert \(W\) to get \(\Theta\))
Now consider \((1, 2)\)-block of KKT conditions:

\[-w_{12} + s_{12} + \lambda \gamma_{12} = 0\]

Because

\[
\begin{bmatrix}
W_{11} & w_{12} \\
w_{21} & w_{22}
\end{bmatrix}
\begin{bmatrix}
\Theta_{11} & \theta_{12} \\
\theta_{21} & \theta_{22}
\end{bmatrix}
= \begin{bmatrix}
I & 0 \\
0 & 1
\end{bmatrix},
\]

we know that

\[w_{12} = -W_{11} \theta_{12} / \theta_{22}.\]

Substituting this into the above,

\[W_{11} \frac{\theta_{12}}{\theta_{22}} + s_{12} + \lambda \gamma_{12} = 0\]

Letting \(\beta = \theta_{12} / \theta_{22}\) and noting that \(\theta_{22} > 0\) at solution, this is

\[W_{11} \beta + s_{12} + \lambda \rho = 0\]

where \(\rho \in \partial \| \beta \|_1\). What does this condition look like?
These are exactly the KKT conditions for

$$\min_{\beta \in \mathbb{R}^{p-1}} \beta^T W_{11} \beta + s^T_{12} \beta + \lambda \|\beta\|_1$$

which is (basically) a lasso problem and can be itself solved quickly via coordinate descent.

From $\beta$ we get $w_{12} = -W_{11} \beta$, and set $w_{21} = w_{12}^T$. Then $\theta_{12}, \theta_{22}$ are obtained from

$$\begin{bmatrix} W_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} \Theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix}, \text{ and}$$

we set $\theta_{21} = \theta_{12}^T$

The next step moves on to a different column of $W$, and so on; hence we have reduced the graphical lasso problem to a repeated sequence of lasso problems.
This coordinate descent approach for the graphical lasso, usually called glasso algorithm (Friedman et al. 2007) is very efficient and scales well.

Meanwhile, people have noticed that using glasso algorithm, it can happen that the objective function doesn’t decrease monotonically across iterations — is this a bug?

No! The glasso algorithm makes a variable transformation and solves in terms of coordinate blocks of $\mathbf{W}$; note that these are not coordinate blocks of original variable $\Theta$, so strictly speaking it is not a coordinate descent algorithm.

However, it can be shown that glasso is doing coordinate ascent on the dual problem (Mazumder et al. 2011)
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\(^2\):

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

where \(\lambda_{\text{max}} = \|X^T y\|_\infty\) (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:

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

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

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

Now consider computing

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

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
(From El Ghaoui et al. 2010)
Another dual argument shows that

\[
\max_{u \in \mathbb{R}^n} X_i^T u \quad \text{subject to} \quad g(u) \geq \gamma
\]

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

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

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
Screening rules for graphical lasso

Graphical lasso computations can be significantly accelerated by using a clever screening rule (this is analogous to the SAFE rules for the lasso)

Mazumder et al. (2011), Witten et al. (2011) examine the KKT conditions:

\[-\Theta^{-1} + S + \lambda \Gamma = 0\]

and conclude that \(\Theta\) is block diagonal over variables \(C_1, C_2\) if and only if \(|S_{ij}| \leq \lambda\) for all \(i \in C_1, j \in C_2\). Why?

- If \(\Theta\) is block diagonal, then so is \(\Theta^{-1}\), and thus \(|S_{ij}| \leq \lambda\) for \(i \in C_1, j \in C_2\)
- If \(|S_{ij}| \leq \lambda\) for \(i \in C_1, j \in C_2\), then the KKT conditions are satisfied with \(\Theta^{-1}\) block diagonal, so \(\Theta\) is block diagonal

Exact same idea extends to multiple blocks. Hence group structure in graphical lasso solution is just given by covariance thresholding.
Early coordinate descent references in statistics and ML:

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

Theory for coordinate descent:

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

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