Coordinate Descent Algorithm

George Michailidis

gmichail@ucla.edu

STAT 102B

Coordinate Descent Algorithm

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex and differentiable function

Coordinate Descent update
At iteration k select coordinate $j_k \in \{1, \cdots, n\}$ and update it according to

$$x_{k+1}(j_k) = x_k(j_k) - \eta_k \nabla_{j_k} f(x_k) \tag{1}$$

where $\nabla_{j_k} f(x_k) \in \mathbb{R}$ denotes the j_k-th coordinate of the gradient vector

In compact mathematical notation

$$x_{k+1} = x_k - \eta_k \nabla_{j_k} f(x_k) e_{j_k} \tag{2}$$

where $e_{j_k}^\top = (0\ 0\ 0 \cdots 1\ 0\ 0\ 0)$ is a vector with all zeros and a single 1 in the j_k position

Illustration of Coordinate Descent - I

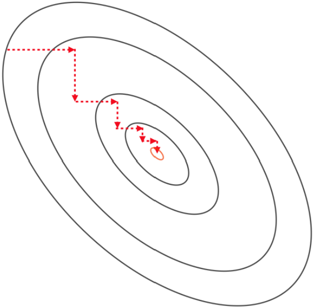

Figure 1: Illustration of Coordinate Descent Algorithm

Illustration of Coordinate Descent - II

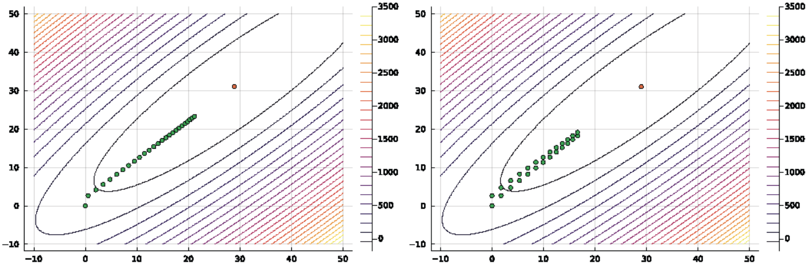

Figure 2: Left panel: GD with step size $\eta = 1/L$; Right panel: CD with step size $\eta = 1/L$, where L is the Lipschitz constant

Remarks

- Coordinate Descent updates one coordinate (parameter) at a time while keeping others fixed
- Efficient when:
  - The objective function decomposes nicely across coordinates; e.g.,

$$f(x) = \sum_{j=1}^{n} f_j(x_j)$$

  - Closed-form updates are available (e.g., in regression problems)
- Used widely for high-dimensional problems in statistics (large number of variables) where full gradient methods can become expensive

Coordinate Descent for Linear Regression - I

Objective function: SSE(β)

$$\min_{\beta} \frac{1}{2n} \|y - X\beta\|_2^2 = \frac{1}{2n} \|y - \sum_{j=1}^{p} \beta_j x_j\|_2^2$$

where $\beta_j \in \mathbb{R}, j = 1, \cdots, p$ and x_j are column vectors of dimension n containing the data for each variable

We would like to calculate the gradient of SSE(β) with respect to a single regression coefficient β_j

Coordinate Descent for Linear Regression - II

Rewrite the SSE(β) function as

$$\mathrm{SSE}(\beta) = \frac{1}{2n}\|y - \sum_{\ell=1, j\neq\ell} \beta_\ell x_\ell - \beta_j x_j\|_2^2$$

Let $r_j = y - \sum_{\ell=1, j\neq\ell} \beta_\ell x_\ell$

This is the partial residual for variable j

Then,

$$\begin{aligned} \mathrm{SSE}(\beta) =& \frac{1}{2n}\|r_j - \beta_j x_j\|_2^2 \\ =& \frac{1}{2n}(r_j - \beta_j x_j)^\top (r_j - \beta_j x_j) \\ =& \frac{1}{2n}(r_j^\top r_j + \beta_j^2 x_j^\top x_j - 2\beta_j r_j^\top x_j) \end{aligned} \tag{3}$$

Coordinate Descent for Linear Regression - III

$$\nabla \mathrm{SSE}_{\beta_j} = \frac{1}{n}(\beta_j x_j^\top x_j - r_j^\top x_j)$$

Since β_j is a scalar, solve the gradient equation to get

$$\nabla \mathrm{SSE}_{\beta_j} = 0 \Longrightarrow \beta_j = \frac{r_j^\top x_j}{x_j^\top x_j}$$

Coordinate Descent for Linear Regression - IV

1. Initialize $\beta_0 \in \mathbb{R}^p$
2. While stopping criterion>tolerance do:
   - For $j = 1, \cdots, p$
     2.1 Calculate the residual $r_j = y - \sum_{\ell=1,\ell\neq j}^{p} \beta_\ell x_\ell$
     2.2 Calculate $\beta_j = \frac{r_j^\top x_j}{x_j^\top x_j}$
   - Calculate the value of the stopping criterion

Coordinate Descent for Ridge Regression

1. Initialize $\beta_0 \in \mathbb{R}^p$
2. While stopping criterion>tolerance do:
   - For $j = 1, \cdots, p$
     2.1 Calculate the residual $r_j = y - \sum_{\ell=1,\ell\neq j}^{p} \beta_\ell x_\ell$
     2.2 Calculate $\beta_j = \frac{r_j^\top x_j}{x_j^\top x_j + \lambda}$
   - Calculate the value of the stopping criterion

Illustration of CD for Linear Regression

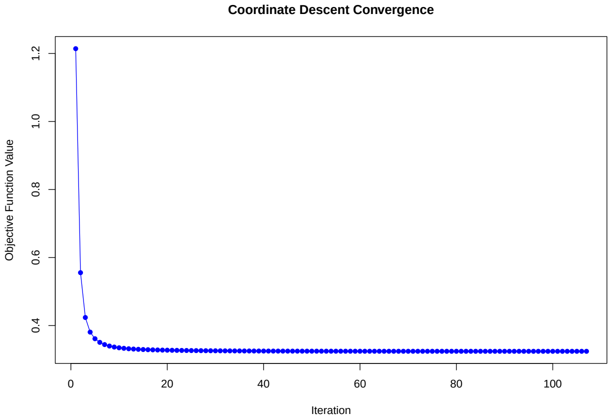

Figure 3: $n = 500, p = 200, CN = 100$

Illustration of CD for Ridge Regression

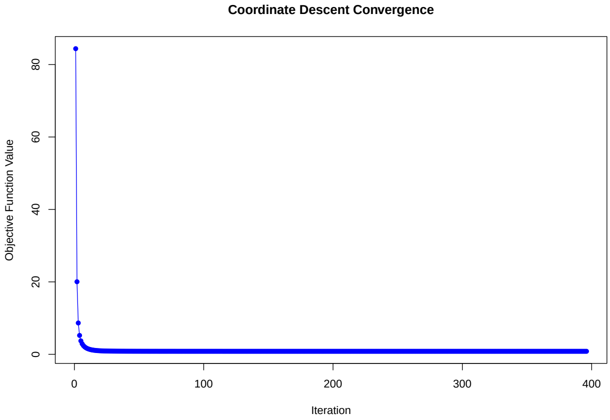

Figure 4: $n = 500, p = 200, CN = 10000, \lambda = 0.2$

Coordinate Descent for Lasso Regression - I

Recall that Lasso uses ℓ_1 regularization

For the update of the j-th regression coefficient, we need to solve the following optimization problem

$$\min_{\beta_j} \frac{1}{2n} \|r_j - \beta_j x_j\|_2^2 + \lambda |\beta_j| = f(\beta) + \lambda g(\beta)$$

Recall that the minimizer of $f(\beta) = \frac{1}{2n} \|r_j - \beta_j x_j\|_2^2$ is given by

$$\beta_j = \frac{r_j^\top x_j}{x_j^\top x_j}$$

Coordinate Descent for Lasso Regression - II

Following the rationale of proximal gradient, let

$$\tilde{\beta}_j = \frac{r_j^\top x_j}{x_j^\top x_j}$$

Then, the update of for β_j is given by

$$\beta_j \leftarrow \text{prox}_{\frac{1}{x_j^\top x_j}, \lambda|\beta_j|}(\tilde{\beta}_j) = \begin{cases} \tilde{\beta}_j - \lambda \frac{1}{x_j^\top x_j}, & \text{if } \tilde{\beta}_j > \lambda \frac{1}{x_j^\top x_j} \\ 0, & \text{if } |\tilde{\beta}_j| \leq \lambda \frac{1}{x_j^\top x_j} \\ \tilde{\beta}_j + \lambda \frac{1}{x_j^\top x_j}, & \text{if } \tilde{\beta}_j < -\lambda \frac{1}{x_j^\top x_j} \end{cases}$$

Coordinate Descent for Lasso Regression - III

The update for β_j can be alternatively written as:

$$\beta_j \leftarrow \frac{1}{x_j^\top x_j} \cdot \mathrm{prox}_{1,\lambda\|\cdot\|_1}(x_j^\top r_j)$$

which is equivalent to:

$$\beta_j \leftarrow \frac{1}{x_j^\top x_j} \cdot \begin{cases} x_j^\top r_j - \lambda, & \text{if } x_j^\top r_j > \lambda \\ 0, & \text{if } |x_j^\top r_j| \leq \lambda \\ x_j^\top r_j + \lambda, & \text{if } x_j^\top r_j < -\lambda \end{cases}$$

Rescaling the Proximal Operator - II

The factor $\frac{1}{a^2}$ is a constant and does not impact the optimization problem
Note that the new optimization problem is with respect to y (since this is the new variable of interest now, not x after the previouschange of variable):

$$\arg\min_y \left\{ \frac{1}{2}\|y - az\|_2^2 + \lambda|y| \right\} = \text{prox}_{1,\lambda}(az) = z^*$$

Then, back-substitute: $x^* = \frac{z^*}{a}$

Final Result:

$$\text{prox}_{\frac{1}{a},\lambda|\cdot|}(v) = \frac{1}{a} \cdot \text{prox}_{1,\lambda}(av)$$

Illustration of CD for Lasso Regression - I

Consider a data set comprising $n = 400$ observations and $p = 500$ predictors

Further, only 5% of the regression coefficients of the predictors are non-zero

Illustration of CD for Lasso Regression - II

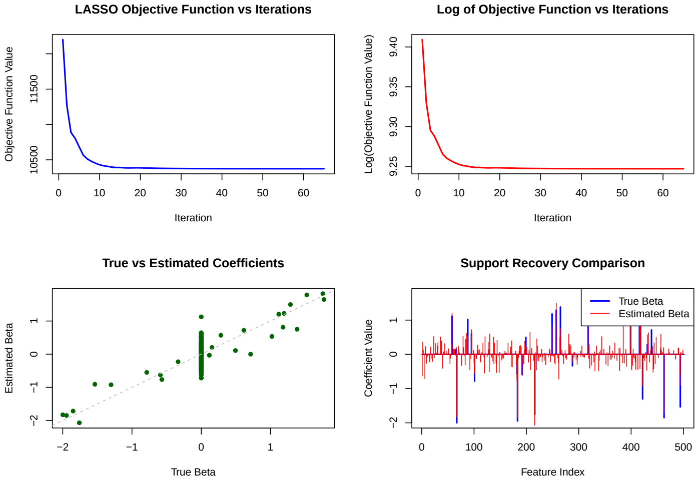

Figure 5: $\lambda = 1$; TP=24, FP=257, TN=218, FN=1

Illustration of CD for Lasso Regression - III

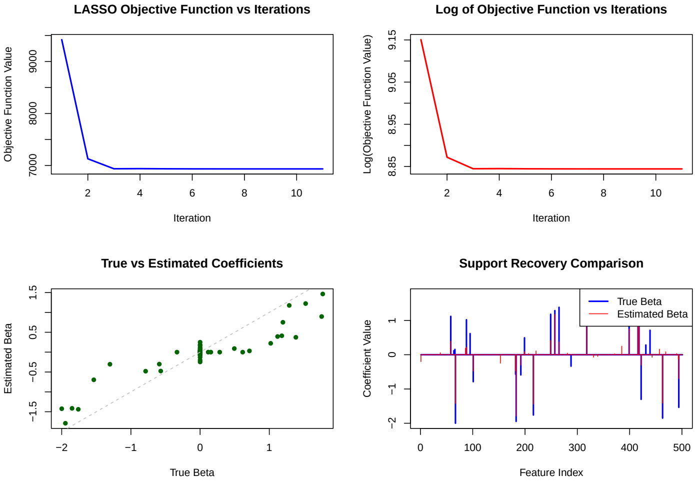

Figure 6: $\lambda = 10$; TP=20, FP=18, TN=457, FN=5

Advantages and Limitations of Coordinate Descent

Advantages:

- Simple implementation
- Memory efficient (no need to store full gradient)
- Works well for large-scale problems
- Very efficient for sparse models
  ideal for ℓ_1 regularization (Lasso)

Limitations:

- Requires additive structure in the coordinates/parameters
- Most effective, if closed form solution of the gradient for each coordinate/parameter is given in closed form