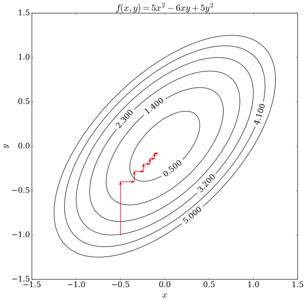
Optimization in Machine Learning

Coordinate descent



Learning goals

- Axes as descent direction
- Convergence criteria of CD
- CD on linear model and Lasso: soft thresholding

COORDINATE DESCENT

- Idea: instead of gradient, use coordinate directions for descent
- No gradient needed: objective function may be non-differentiable

Approach:

- Select starting point $\boldsymbol{x}^{[0]} = (x_1^{[0]}, \ldots, x_d^{[0]})$
- Step t: Minimize f along x_i for each dimension i for fixed $x_1^{[t]}, \ldots, x_{i-1}^{[t]}$ and $x_{i+1}^{[t-1]}, \ldots, x_d^{[t-1]}$
- Minimum is determined with (exact/inexact) line search
- Order of dimensions: cyclic $1, 2, \ldots, d$, or any other permutation

CONVERGENCE CRITERIA

- CD stops at a point that is minimal along each coordinate axis; is this a global minimizer? ▸ Click for source

  Case 1: f convex and differentiable $\Rightarrow$ **yes**

  Case 2: f convex, not differentiable $\Rightarrow$ **no** in general (next slide)

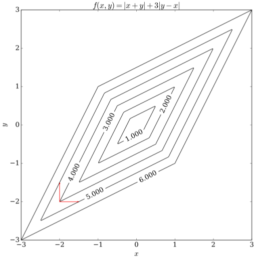
  Case 3: f = convex differentiable + **convex separable** $\Rightarrow$ **yes**

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + \sum_{i=1}^{d} h_i(x_i)$$

with g convex differentiable and h_i convex, possibly nonsmooth, univariate functions

EXAMPLE: NON-DIFFERENTIABLE FUNCTION

- No general convergence for convex, non-differentiable functions (case 2)
- Counterexample $f(x, y) = |x + y| + 3|y - x|$, minimum at $(0, 0)$
- At the red point, a step along either axis increases f, although both steps together would bring us closer to the optimum
- CD gets stuck at a point that is not the minimum



▸ Click for source

EXAMPLE: LINEAR REGRESSION

- Minimize LM with L2 loss via CD (case 1):

$$\min g(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i=1}^{n} (y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)})^2 = \min_{\boldsymbol{\theta}} \frac{1}{2} ||\mathbf{y} - \mathbf{X}\boldsymbol{\theta}||^2$$

where $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n \times d}$ with columns $\mathbf{x}_1, \dots, \mathbf{x}_d \in \mathbb{R}^n$

Note: smooth toy example where the minimizer along each axis is computed in closed form

- Assume orthogonal design for intuition, i.e., $\mathbf{X}^T\mathbf{X} = \mathbf{I}$:

$$\begin{aligned} g(\boldsymbol{\theta}) &= \frac{1}{2}\mathbf{y}^T\mathbf{y} + \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} - \mathbf{y}^T\mathbf{X}\boldsymbol{\theta} \\ &\overset{(*)}{=} \frac{1}{2}\mathbf{y}^T\mathbf{y} + \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} - \mathbf{y}^T \sum_{k=1}^{d} \mathbf{x}_k \theta_k \end{aligned}$$

$^{(*)}$ $\mathbf{X}\boldsymbol{\theta} = \mathbf{x}_1\theta_1 + \mathbf{x}_2\theta_2 + \cdots + \mathbf{x}_d\theta_d = \sum_{k=1}^{d} \mathbf{x}_k\theta_k$

EXAMPLE: LINEAR REGRESSION

- Exact CD update in direction j:

$$\frac{\partial g(\boldsymbol{\theta})}{\partial \theta_j} = \theta_j - \boldsymbol{y}^T \boldsymbol{x}_j$$

- By solving $\frac{\partial g(\boldsymbol{\theta})}{\partial \theta_j} = 0$ we get

$$\theta_j^* = \boldsymbol{y}^T \boldsymbol{x}_j$$

- Repeat this update for all θ_j

EXAMPLE: SOFT THRESHOLDING

- Minimize LM with L2 loss + L1 regularization via CD (case 3):

$$\min_{\boldsymbol{\theta}} h(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \frac{1}{2}||\boldsymbol{y} - \mathbf{X}\boldsymbol{\theta}||^2 + \lambda||\boldsymbol{\theta}||_1$$

- Again we assume orthogonal design, i.e., $\mathbf{X}^T\mathbf{X} = \boldsymbol{I}$, and can write

$$h(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{y}^T\boldsymbol{y} + \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} - \sum_{k=1}^{d} \boldsymbol{y}^T\boldsymbol{x}_k\theta_k + \lambda\sum_{k=1}^{d}|\theta_k|$$

- Since $|\cdot|$ not differentiable at 0, distinguish three cases (see next slide)

EXAMPLE: SOFT THRESHOLDING

- **Case 1:** $\theta_j > 0$. Then $|\theta_j| = \theta_j$ and

$$0 = \frac{\partial h(\boldsymbol{\theta})}{\partial \theta_j} = \theta_j - \boldsymbol{y}^T \boldsymbol{x}_j + \lambda \quad \Leftrightarrow \quad \theta^*_{j,\text{Lasso}} = \theta^*_j - \lambda$$

Consistent with $\theta_j > 0$ only if $\theta^*_j > \lambda$

- **Case 2:** $\theta_j < 0$. Then $|\theta_j| = -\theta_j$ and

$$0 = \frac{\partial h(\boldsymbol{\theta})}{\partial \theta_j} = \theta_j - \boldsymbol{y}^T \boldsymbol{x}_j - \lambda \quad \Leftrightarrow \quad \theta^*_{j,\text{Lasso}} = \theta^*_j + \lambda$$

Consistent with $\theta_j < 0$ only if $\theta^*_j < -\lambda$

- **Case 3:** $\theta_j = 0$, remaining case where $|\theta^*_j| \leq \lambda$

EXAMPLE: SOFT THRESHOLDING

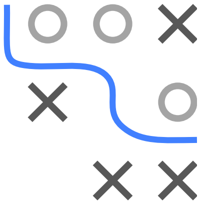
We can write the solution as:

$$\theta^*_{j,\text{Lasso}} = \begin{cases} \theta^*_j - \lambda & \text{if } \theta^*_j > \lambda \\ \theta^*_j + \lambda & \text{if } \theta^*_j < -\lambda \\ 0 & \text{if } \theta^*_j \in [-\lambda, \lambda] \end{cases}$$

This operation is called soft thresholding: Coefs for which solution to unregularized problem is not larger than threshold ($|\theta^*_j| \leq \lambda$) are shrunk to zero

NB: Rigorous derivation of soft thresholding operator not trivial (subgradients)

CD FOR STATISTICS AND ML



Why is it being used?

- Easy to implement
- Scalable: no storage/operations on large objects, just current point
  $\Rightarrow$ Good implementation can achieve state-of-the-art performance
- Applicable for non-differentiable (but convex separable) objectives

Examples:

- Lasso regression, Lasso GLM, graphical Lasso
- Support Vector Machines
- Regression with non-convex penalties

Limitations:

- Slow (zig-zagging) if coordinates strongly correlated (title figure)