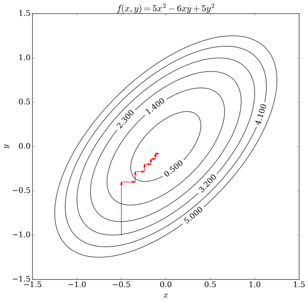
Optimization in Machine Learning

Coordinate descent



Learning goals

- Axes as descent direction
- CD on linear model and LASSO
- Soft-Thresholding operator

COORDINATE DESCENT

If derivative of objective function does not exist / is unknown we cannot compute a descent direction analytically and an **inexact** procedure must be used.

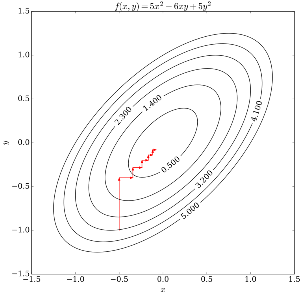
Idea: Use direction of coordinate axes as "descent directions".

In the simplest case we run iteratively over all coordinates $\{1, ..., d\}$ and minimize f with respect to the corresponding dimension.

COORDINATE DESCENT

- First a starting point $\boldsymbol{x}^{[0]} = \left(x_1^{[0]}, \ldots, x_d^{[0]}\right)$ is selected.
- In step t we search the value x_i for each dimension $i \in \{1, 2, \ldots, d\}$ that minimizes f, given $x_1^{[t]}, \ldots, x_{i-1}^{[t]}$ and $x_{i-1}^{[t-1]}, \ldots, x_d^{[t-1]}$:

$$
\begin{aligned}
x_1^{[t]} &= \arg\min_{x_1} f(x_1, x_2^{[t-1]}, x_3^{[t-1]}, \ldots, x_d^{[t-1]}) \\
x_2^{[t]} &= \arg\min_{x_2} f(x_1^{[t]}, x_2, x_3^{[t-1]}, \ldots, x_d^{[t-1]}) \\
x_3^{[t]} &= \arg\min_{x_3} f(x_1^{[t]}, x_2^{[t]}, x_3, \ldots, x_d^{[t-1]}) \\
&\vdots \\
x_d^{[t]} &= \arg\min_{x_d} f(x_1^{[t]}, x_2^{[t]}, x_3^{[t]}, \ldots, x_d)
\end{aligned}
$$



`https://commons.wikimedia.org/wiki/File:Coordinate_descent.svg`

COORDINATE DESCENT

- Minimum is determined with (exact / inexact) line search
- Order in which the dimensions are gone through can be any permutation of $\{1, 2, \ldots, d\}$
- **Convergence:** if $f(\cdot)$ is continuously differentiable and the univariate minimization problems have unique solutions, the sequence $\boldsymbol{x}^{[t]}$ converges to $\boldsymbol{x}^*$ with $\nabla f(\boldsymbol{x}^*) = 0$.

  The following holds:

$$f(\boldsymbol{x}^{[0]}) \geq f(\boldsymbol{x}^{[1]}) \geq f(\boldsymbol{x}^{[2]}) \geq \ldots$$

EXAMPLE: LINEAR REGRESSION

Minimize LM with L2-loss via CD:

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

where $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ with columns $\boldsymbol{X}_1, \dots, \boldsymbol{X}_p \in \mathbb{R}^n$.

Assumption: data is scaled $\mathbf{X}_j^\top \mathbf{1} = 0$ and $\mathbf{X}^\top \mathbf{X} = \mathbf{1}_p$.

g simplifies to

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

$^{(*)}$ $\mathbf{X}\boldsymbol{\theta} = \mathbf{X}_1\boldsymbol{\theta}_1 + \mathbf{X}_2\boldsymbol{\theta}_2 + ... + \mathbf{X}_p\boldsymbol{\theta}_p = \sum_{k=1}^{p} \mathbf{X}_k \boldsymbol{\theta}_k$.

EXAMPLE: LINEAR REGRESSION

To compute the exact CD update in direction j we compute

$$\frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_j} \quad = \quad \boldsymbol{\theta}_j - \mathbf{y}^\top \mathbf{X}_j$$

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

$$\boldsymbol{\theta}_j^* = \mathbf{y}^\top \mathbf{X}_j$$

as exact update for CD in direction j. We repeat this update over all variables.

SOFT THRESHOLDING OPERATOR

Minimize LM with L2-loss and L1 regularization via CD:

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

We can write $h(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{y}^\top\boldsymbol{y} + \frac{1}{2}\boldsymbol{\theta}^\top\boldsymbol{\theta} - \sum_{k=1}^{p}(\boldsymbol{y}^\top\mathbf{X}_k\boldsymbol{\theta}_k + \lambda|\boldsymbol{\theta}_k|)$.

Because $|\cdot|$ is not differentiable, we distinguish three cases:

- **Case 1**: $\boldsymbol{\theta}_j > 0$. Then $|\boldsymbol{\theta}_j| = \boldsymbol{\theta}_j$ and

$$0 \quad = \quad \frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_j} = \boldsymbol{\theta}_j - \boldsymbol{y}^\top\mathbf{X}_j + \lambda \qquad \Leftrightarrow \qquad \boldsymbol{\theta}^*_{j,\text{LASSO}} = \boldsymbol{\theta}^*_j - \lambda$$

- **Case 2**: $\boldsymbol{\theta}_j < 0$. Then $|\boldsymbol{\theta}_j| = -\boldsymbol{\theta}_j$ and

$$0 \quad = \quad \frac{\partial g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_j} = \boldsymbol{\theta}_j - \boldsymbol{y}^\top\mathbf{X}_j - \lambda \qquad \Leftrightarrow \qquad \boldsymbol{\theta}^*_{j,\text{LASSO}} = \boldsymbol{\theta}^*_j + \lambda$$

- **Case 3**: $\boldsymbol{\theta}_j = 0$.

SOFT THRESHOLDING OPERATOR

We can write the solution as:

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

which is also referred to as **soft-thresholding operator**. Coefficients for which the solution to the unregularized problem are smaller than a threshold, $|\boldsymbol{\theta}^*_j| < \lambda$, are shrinked to zero.

Note:

- For case 1, we require

$$\boldsymbol{\theta}^*_{j,\text{LASSO}} = \boldsymbol{\theta}^*_j - \lambda > 0 \qquad \Leftrightarrow \qquad \boldsymbol{\theta}^*_j > \lambda$$

- For case 2, we require

$$\boldsymbol{\theta}^*_{j,\text{LASSO}} = \boldsymbol{\theta}^*_j + \lambda < 0 \qquad \Leftrightarrow \qquad \boldsymbol{\theta}^*_j < -\lambda$$

CD FOR STATISTICS AND ML

Why is it being used?

- Very easy to implement.
- Good implementation can achieve state-of-the-art performance.
- Scalable, e.g. no storage or operations on large objects, only the current point
- Applicable in both differentiable and derivative-free cases.

Examples:

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