Find the Lasso Solution

He He

CDS, NYU

Feb 16, 2021

Quadratic Programming

How to find the Lasso solution?

- How to solve the Lasso?

$$\min_{w \in \mathrm{R}^d} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2 + \lambda \|w\|_1$$

- $\|w\|_1 = |w_1| + |w_2|$ is not differentiable!

Rewrite the Absolute Function

- Consider any number $a \in \mathbb{R}$.
- Let the **positive part** of a be
$$a^{+} = a\mathbb{1}(a \geqslant 0).$$
- Let the **negative part** of a be
$$a^{-} = -a\mathbb{1}(a \leqslant 0).$$
- Do you see why $a^{+} \geqslant 0$ and $a^{-} \geqslant 0$?
- How do you write a in terms of a^{+} and a^{-}?
- How do you write $|a|$ in terms of a^{+} and a^{-}?

The Lasso as a Quadratic Program

We will show: substituting $w = w^+ - w^-$ and $|w| = w^+ + w^-$ gives an equivalent problem:

$$\min_{w^+, w^-} \quad \sum_{i=1}^{n} \left(\left(w^+ - w^-\right)^T x_i - y_i \right)^2 + \lambda 1^T \left(w^+ + w^-\right)$$

$$\text{subject to} \quad w_i^+ \geqslant 0 \text{ for all } i \qquad w_i^- \geqslant 0 \text{ for all } i,$$

- Objective is **differentiable** (in fact, **convex and quadratic**)
- $2d$ variables vs d variables and $2d$ constraints vs no constraints
- A "**quadratic program**": a convex quadratic objective with linear constraints.
  - Could plug this into a generic QP solver.

Possible point of confusion

We have claimed that this objective is equivalent to lasso problem:

$$\begin{aligned}\min_{w^+,w^-} \quad & \sum_{i=1}^{n}\left(\left(w^+-w^-\right)^T x_i-y_i\right)^2+\lambda 1^T\left(w^++w^-\right) \\ \text{subject to} \quad & w_i^+\geqslant 0 \text{ for all } i \qquad w_i^-\geqslant 0 \text{ for all } i,\end{aligned}$$

- When we plug this optimization problem into a QP solver,
  - it just sees $2d$ variables and $2d$ constraints.
  - Doesn't know we want w_i^+ and w_i^- to be positive and negative parts of w_i.
- Turns out – they will come out that way as a result of the optimization!
- But to eliminate confusion, let's start by calling them a_i and b_i and prove our claim...

The Lasso as a Quadratic Program

Lasso problem is trivially equivalent to the following:

$$
\begin{aligned}
\min_{w} \min_{a,b} \quad & \sum_{i=1}^{n} \left((a-b)^T x_i - y_i \right)^2 + \lambda 1^T (a+b) \\
\text{subject to} \quad & a_i \geqslant 0 \text{ for all } i \qquad b_i \geqslant 0 \text{ for all } i, \\
& a - b = w \\
& a + b = |w|
\end{aligned}
$$

Claim: Don't need constraint $a + b = |w|$.

Exercise: rove by showing that the optimal solutions a^* and b^* satisfies $\min(a^*, b^*) = 0$, hence $a^* + b^* = |w|$.

The Lasso as a Quadratic Program

$$
\begin{aligned}
\min_{w} \min_{a,b} \quad & \sum_{i=1}^{n} \left((a-b)^T x_i - y_i \right)^2 + \lambda 1^T (a+b) \\
\text{subject to} \quad & a_i \geqslant 0 \text{ for all } i \qquad b_i \geqslant 0 \text{ for all } i, \\
& a - b = w
\end{aligned}
$$

Claim: Can remove $\min_w$ and the constraint $a - b = w$.

Exercise: Prove by switching the order of the minimization.

Projected SGD

Now the objective is differentiable, but how do we handle the constraints?

$$
\begin{aligned}
\min_{w^+, w^- \in \mathbb{R}^d} & \sum_{i=1}^{n} \left(\left(w^+ - w^-\right)^T x_i - y_i \right)^2 + \lambda 1^T \left(w^+ + w^-\right) \\
\text{subject to } & w_i^+ \geqslant 0 \text{ for all } i \\
& w_i^- \geqslant 0 \text{ for all } i
\end{aligned}
$$

- Just like SGD, but after each step
  - Project w^+ and w^- into the constraint set.
  - In other words, if any component of w^+ or w^- becomes negative, set it back to 0.

Coordinate Descent (Shooting Method)

Coordinate Descent Method

Goal: Minimize $L(w) = L(w_1, \ldots, w_d)$ over $w = (w_1, \ldots, w_d) \in \mathrm{R}^d$.

In gradient descent or SGD, each step potentially changes all entries of w.

In **coordinate descent**, each step adjusts only a single coordinate w_i.

$$w_i^{\text{new}} = \underset{w_i}{\arg\min}\, L(w_1, \ldots, w_{i-1}, \mathrm{w_i}, w_{i+1}, \ldots, w_d)$$

- Solving this argmin may itself be an iterative process.
- Coordinate descent is great when it's easy or easier to minimize w.r.t. one coordinate at a time

Coordinate Descent Method

Goal: Minimize $L(w) = L(w_1, \ldots w_d)$ over $w = (w_1, \ldots, w_d) \in \mathrm{R}^d$.

- **Initialize** $w^{(0)} = 0$
- **while** not converged:
  - Choose a coordinate $j \in \{1, \ldots, d\}$
  - $w_j^{\text{new}} \leftarrow \arg\min_{w_j} L(w_1^{(t)}, \ldots, w_{j-1}^{(t)}, \mathrm{w_j}, w_{j+1}^{(t)}, \ldots, w_d^{(t)})$
  - $w_j^{(t+1)} \leftarrow w_j^{\text{new}}$ and $w^{(t+1)} \leftarrow w^{(t)}$
  - $t \leftarrow t + 1$

- Random coordinate choice $\implies$ **stochastic coordinate descent**
- Cyclic coordinate choice $\implies$ **cyclic coordinate descent**

In general, we will adjust each coordinate several times.

Coordinate Descent Method for Lasso

- Why mention coordinate descent for Lasso?
- In Lasso, the coordinate minimization has a **closed form solution**!

Coordinate Descent Method for Lasso

Closed Form Coordinate Minimization for Lasso

$$\hat{w}_j = \underset{w_j \in \mathbb{R}}{\arg\min} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2 + \lambda |w|_1$$

Then

$$\hat{w}_j = \begin{cases} (c_j + \lambda)/a_j & \text{if } c_j < -\lambda \\ 0 & \text{if } c_j \in [-\lambda, \lambda] \\ (c_j - \lambda)/a_j & \text{if } c_j > \lambda \end{cases}$$

$$a_j = 2\sum_{i=1}^{n} x_{i,j}^2 \qquad c_j = 2\sum_{i=1}^{n} x_{i,j}(y_i - w_{-j}^T x_{i,-j})$$

where w_{-j} is w without component j and similarly for $x_{i,-j}$.

Coordinate Descent in General

- Theoretically, coordinate descent is not competitive, e.g. its convergence rate is slower than GD and the iteration cost is similar
- But it works very well for certain problems
- Very simple and easy to implement
- Example applications: lasso regression, SVMs