Reduction from Mean-Variance to ReHLine

aorazalin

July 2024

1 Introduction

Recall that the python solver class `ReHLineLinear` was implemented to solve the `ReHLine` optimization with the addition of a linear term. Luckily, precisely similar rehline algorithm can be used to solve the extended problem. A natural way to implement this would be to create a separate internal solver in C++, say `rehline_linear_solver` and a wrapper solver python class `ReHLineLinear` that would call this internal solver. Note that, however, the `ReHLineLinear` python class in my implementation didn't incorporate any additional C++ internal solver and solely used the original internal solver (so I haven't wrote any C++ code). This section will explain how this was done.

2 Derivation

Given the original `ReHLineLinear`$(\mathbf{U}, \mathbf{V}, \mathbf{S}, \mathbf{T}, \mathbf{A}, \mathbf{b}, \mu)$ problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} \mathcal{P}^{linear}(\mathbf{w}) \text{ s.t. } \mathbf{A}\mathbf{w} + \mathbf{b} \geq \mathbf{0}$$

where

$$\mathcal{P}^{linear}(\mathbf{w}) := \frac{1}{2}\mathbf{w}^T\mathbf{w} - \mu^T\mathbf{w} + \sum_{i=1}^{n}\sum_{l=1}^{L} \mathrm{ReLU}(u_{li}\mathbf{w}^T\mathbf{x_i} + v_{li}) + \sum_{i=1}^{n}\sum_{h=1}^{H} \mathrm{ReHU}_{\tau_{hi}}(s_{hi}\mathbf{w}^T\mathbf{x_i} + t_{hi})$$

one can transform this problem into `ReHLine`$(\mathbf{U}, \mathbf{V}', \mathbf{S}, \mathbf{T}', \mathbf{A}, \mathbf{b}')$ by completing the square for $\frac{1}{2}\mathbf{w}^T\mathbf{w} - \mu^T\mathbf{w}$ and shifting $\mathbf{w} \leftarrow \mathbf{w} - \mu$:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \mathcal{P}^{original}(\mathbf{w}) - \frac{1}{2}\mu^T\mu, \text{ s.t. } \mathbf{A}\mathbf{w} + \mathbf{b}' \geq \mathbf{0}$$

where

$$\mathcal{P}^{original}(\mathbf{w}) := \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=1}^{n}\sum_{l=1}^{L} \mathrm{ReLU}(u_{li}\mathbf{w}^T\mathbf{x_i} + v'_{li}) + \sum_{i=1}^{n}\sum_{h=1}^{H} \mathrm{ReHU}_{\tau_{hi}}(s_{hi}\mathbf{w}^T\mathbf{x_i} + t'_{hi})$$

where parameters of the optimizer are shifted as $v'_{li} = v_{li} + u_{li}\mu^T\mathbf{x}_i$, $t'_{hi} = t_{hi} + s_{hi}\mu^T\mathbf{x}_i$, $\mathbf{b}' = \mathbf{b} + \mathbf{A}\mu$ (*).

Thus, `ReHLineLinear`$(\mathbf{U}, \mathbf{V}, \mathbf{S}, \mathbf{T}, \mathbf{A}, \mathbf{b}, \mu)$ can be solved like this:

1. Shift parameters $\mathbf{V}, \mathbf{T}, \mathbf{b} \rightarrow \mathbf{V}', \mathbf{T}', \mathbf{b}'$
2. Solve $\mathbf{w}^o$, ξ^o, $\mathbf{\Lambda}^o$, $\mathbf{\Gamma}^o$, $\mathcal{P}^o$, $\mathcal{D}^o \leftarrow$ `ReHLine`$(\mathbf{U}, \mathbf{V}', \mathbf{S}, \mathbf{T}', \mathbf{A}, \mathbf{b}')$
3. Un-shift results
   - $\mathbf{w} = \mathbf{w}^o + \mu$
   - $\xi, \mathbf{\Lambda}, \mathbf{\Gamma} = \xi^o, \mathbf{\Lambda}^o, \mathbf{\Gamma}^o$
   - $\mathcal{P} = \mathcal{P}^o - \frac{1}{2}\mu^T\mu$
   - $\mathcal{D} = \mathcal{D}^o + \frac{1}{2}\mu^T\mu$

In fact, the algorithm above is precisely equal to running coordinate descent on the `ReHLineLinear` problem.

Apart from what we covered, we haven't only covered why dual variables don't change when we un-shift back to the original problem. It is pretty clear once we look at dual objective functions of both problems:

$$\begin{aligned}\mathcal{D}^{linear}(\xi, \mathbf{\Lambda}, \mathbf{\Gamma}) :=& \frac{1}{2}||\mathbf{A}^T\xi + \mu - \bar{\mathbf{U}}_{(\mathbf{3})}\text{vec}(\mathbf{\Lambda}) - \bar{\mathbf{S}}_{(\mathbf{3})}\text{vec}(\mathbf{\Gamma})||_2^2 + \frac{1}{2}\text{vec}(\mathbf{\Gamma})^T\text{vec}(\mathbf{\Gamma}) \\ &+ \xi^T\mathbf{b} - Tr(\mathbf{\Lambda}^T\mathbf{V}) - Tr(\mathbf{\Gamma}^T\mathbf{T})\end{aligned}$$

and

$$\begin{aligned}\mathcal{D}^{original}(\xi, \mathbf{\Lambda}, \mathbf{\Gamma}) :=& \frac{1}{2}||\mathbf{A}^T\xi - \bar{\mathbf{U}}_{(\mathbf{3})}\text{vec}(\mathbf{\Lambda}) - \bar{\mathbf{S}}_{(\mathbf{3})}\text{vec}(\mathbf{\Gamma})||_2^2 + \frac{1}{2}\text{vec}(\mathbf{\Gamma})^T\text{vec}(\mathbf{\Gamma}) \\ &+ \xi^T\mathbf{b} - Tr(\mathbf{\Lambda}^T\mathbf{V}) - Tr(\mathbf{\Gamma}^T\mathbf{T})\end{aligned}$$

by shifting parameters $\mathbf{T}, \mathbf{V}, \mathbf{b}$ as above one can easily show

$$\mathcal{D}^{linear}(\xi, \mathbf{\Lambda}, \mathbf{\Gamma} \mid \mathbf{A}, \mathbf{U}, \mathbf{V}, \mathbf{S}, \mathbf{T}, \mathbf{A}, \mathbf{b}, \mu) = \mathcal{D}^{original}(\xi, \mathbf{\Lambda}, \mathbf{\Gamma} \mid \mathbf{A}, \mathbf{U}, \mathbf{V}', \mathbf{S}, \mathbf{T}', \mathbf{A}, \mathbf{b}') + \frac{1}{2}\mu^T\mu$$

where $\mathbf{T}', \mathbf{V}', \mathbf{b}'$ are defined as in (*).