Using the NonsmoothPath package

Qiuyu Gu

2022-09-14

The `NonsmoothPath` package contains the implementation of the path-following algorithm to solve case weight adjusted problem

$$(\beta_{0,w}, \beta_w) = \text{argmin}_{\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p} \sum_{i \neq i^*} f(g_i(\beta_0, \beta)) + w f(g_{i^*}(\beta_0, \beta)) + \frac{\lambda}{2} \|\beta\|_2^2, \tag{1}$$

where $g_i(\beta_0, \beta) = a_i \beta_0 + b_i^\top \beta + c_i$ for some $a_i, c_i \in \mathbb{R}, b_i \in \mathbb{R}^p$, and f is a piecewise linear function with one knot at 0, i.e. $f(r) = \alpha_0 \max(r, 0) + \alpha_1 \max(-r, 0)$ for some $\alpha_0 \geq 0, \alpha_1 > 0$. The term $f(g_i(\beta_0, \beta))$ can be viewed as the loss function for i^{th} case. If we consider the quantile regression with the parameter τ, then

$$a_i = -1, b_i = -x_i, c_i = y_i, \alpha_0 = \tau, \alpha_1 = 1 - \tau.$$

And if we consider SVM, then

$$a_i = y_i, b_i = y_i x_i, c_i = -1, \alpha_0 = 0, \alpha_1 = 1.$$

The solution when $w = 1$ corresponds to the full data solution and we assume it is available to us. The algorithm computes the exact solution path of (1), which is shown to be piecewise linear in w, as the parameter w changes from 1 to 0.

Using `NonsmoothPath` functions

For each given data and specified model, function `nonsmooth_path` in `NonsmoothPath` package computes solution path of (1) and outputs the

1. breakpoints,

2. β_0, β and dual variable (θ) at these breakpoints.

Next, We use quantile regression as an example to show how to apply `nonsmooth_path` function to compute solution path for case weight adjusted problem (1). First we generate data from the standard linear model

$$y_i = x_i^\top \beta + \epsilon_i,$$

where covariates $\{x_{ij} : i = 1, \cdots, n,\ j = 1, \cdots, p\}$, coefficients $\{\beta_1, \cdots, \beta_p]\}$ and errors $\{\epsilon_i : i = 1, \cdots, n\}$ are independently generated from standard normal distribution. Here tau is the quantile parameter in quantile regression and lam is the regularization parameter for L_2 penalty.

```
# n = 100
# p = 200
# lam = 50
# tau = 0.1
# beta_true = rnorm(p)
```

```r
# X = matrix(0, ncol = p, nrow = n)
# for (i in 1:n)
#   X[i, ] = rnorm(p)
# eps = rnorm(n)
# Y = X %*% beta_true + eps
```

To compute solution path for the generated data under quantile regression, we can call the function `nonsmooth_path`. All we need to do is to specify the `class = "quantile"` and the value for `obs_index` argument, which is the index of the case we attach parameter w.

```r
# obs_index = 1
# quantile_path = nonsmooth_path(X, Y, lam, obs_index, class = "quantile", tau = tau)
```

The function outputs a list, which contains all the breakpoints and corresponding β_0, β and θ. We can access these values through

```r
# w_vec = quantile_path$W_vec
# beta_0_vec = quantile_path$Beta_0
# beta_mat = quantile_path$Beta
# theta_mat = quantile_path$Theta
```

To make it work for support vector machine (svm), the following changes are needed:

1. Set `class = "svm"`

2. Do not need to specify values for tau.