1 Computing the Lasso Solution

1.1 Proximal Algorithm : General Solution

A convenient algorithm is given by the FISTA (see Beck and Teboulle (2014) for a clear exposition). Assume you want to minimize the following objective function in $x \in \mathbb{R}^p$:

$$f(x) = g(x) + h(x) \tag{1}$$

For which g is convex and differentiable, and h is closed with inexpensive proximal operator. Then do the following:
Set starting values $x^{(0)} = x^{(-1)}$ and for $k \geq 1$ repeat:

(1) $y = x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-1)} - x^{(k-2)})$.

(2) $x^{(k)} = \text{prox}_{t_k h}\left(y - t_k \bigtriangledown g(y)\right)$.

(3) Increase k by one unit.

Use $t_k = \frac{1}{L}$ where L is the Lipschitz constant for the gradient $\bigtriangledown g(.)$ or determine t_k by line search. The proximal operator is given by:

$$\text{prox}_h(x) = \arg\min_u \left(h(u) + \frac{1}{2}\|u - x\|_2^2 \right) \tag{2}$$

1.2 Application to Solve the Lasso

For example, this method can be used to resolve the Lasso problem:

$$\min_\beta \frac{1}{n} \sum_{i=1}^n (y_i - x_i^T \beta)^2 + \lambda \|\beta\|_1 \tag{3}$$

In this case, $L = \frac{2}{n}\lambda_{\max}\left(X^T X\right)$ and $\text{prox}_h(x) = (x - \lambda)_+ \text{sign}(x)$. $\lambda_{\max}(A)$ denotes the maximal eigenvalue of a matrix A. Note that the proximal operator can also be defined in this case as: $\text{prox}_h(x) = (x-\lambda)_+ - (-x-\lambda)_+$. If L is unknown, one can use the backtracking line search procedure.

We finally give the FISTA algorithm for the Lasso in a readily usable form. Set for initialization $\beta_{(1)} = \alpha_{(1)} = 0_p$, $\theta_{(1)} = 1$ and $\eta = \left(\frac{2}{n}\lambda_{\max}\left(X^T X\right)\right)^{-1}$. Repeat until convergence of the sequence $\beta^{(1)}, \beta^{(2)}, \ldots$

(1) $\theta_{(k+1)} = \left(1 + \sqrt{1 + 4\theta_{(k)}^2}\right)/2$ and set $\delta_{(k)} = (1 - \theta_{(k)})/\theta_{(k+1)}$.

(2) $\beta^{(k+1)} = \text{prox}_{\eta\lambda}\left(\alpha_{(k)} - \eta \bigtriangledown g(\alpha_{(k)})\right)$

(3) $\alpha_{(k+1)} = (1 - \delta_{(k)})\beta^{(k+1)} + \delta_{(k)}\beta^{(k)}$.

(4) Increase k by one unit.

Here : $\bigtriangledown g(\beta) = -\frac{2}{n}\sum_{i=1}^n (y_i - x_i^T\beta)x_i$.

2 Bibliography

Beck, A. and Teboulle, M. (2014): “A fast dual proximal gradient algorithm for convex minimization and applications”. *Operations Research Letters*, 42(1):1 – 6.