Computational issues

Chunlong Luo
whu_boom@163.com

June 29, 2016

Computational method

Indroduction

To estimate the parameters $\boldsymbol{\beta_0}$ and $\boldsymbol{\gamma}$,the penalized log-likelihood $l_{\zeta_1,\zeta_2}(\boldsymbol{\beta_0},\boldsymbol{\gamma})$ can be formulated as

$$(\hat{\boldsymbol{\beta}}_{\mathbf{0}},\hat{\boldsymbol{\gamma}}) = \underset{\boldsymbol{\beta_0},\boldsymbol{\gamma}}{\operatorname{argmin}} \Big(-l(\boldsymbol{\beta_0},\boldsymbol{\gamma}) + \zeta_1 \boldsymbol{J_1}(\boldsymbol{\beta_0}) + \zeta_2 \boldsymbol{J_2}(\boldsymbol{\gamma}) \Big) \quad (1)$$

- The alogrithm for solving problem is based on proximal gradient algorithms.
- Proximal gradient method belongs to a class of algorithms, called proximal algorithms, for solving convex optimization problems.
- The base operation of proximal algorithms is evaluating the proximal operator of a function, which itself involves solving a small convex optimization problem.

Proximal operator

Let $f : \boldsymbol{R}^n \to \boldsymbol{R} \cup \{+\infty\}$ be a closed proper convex function,the proximal operator $\mathbf{prox}_f : \boldsymbol{R}^n \to \boldsymbol{R}^n$ of f is defined by

$$\mathbf{prox}_f(v) = \underset{x}{\operatorname{argmin}} \left(f(x) + (1/2)\|x - v\|_2^2 \right) \tag{2}$$

where $\|\cdot\|_2$ is L_2 norm.

Interpretations

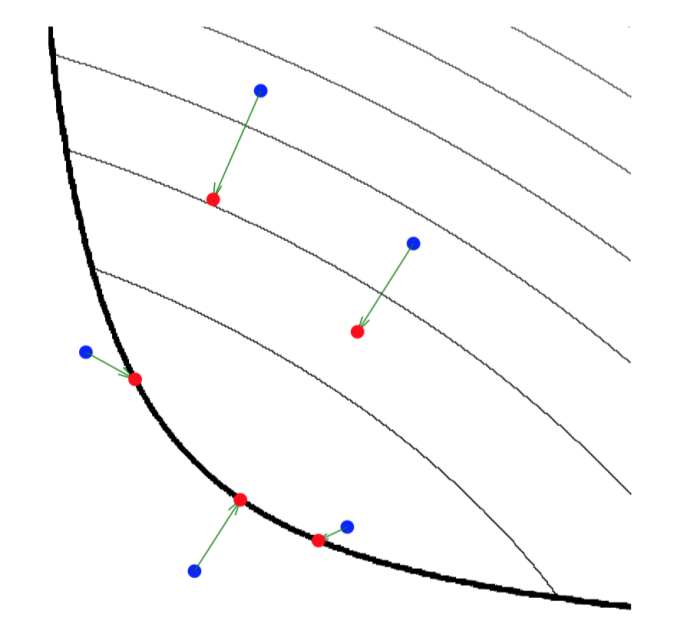

Figure 1: Evaluating a proximal operator at various points.

Figure 1 depicts what a proximal operator does. The points move towards the minimum of the function.The definition indicates that $\mathbf{prox}_f(v)$ is a point that compromises between minimizing f and being near to v.

Interpretations

The proximal operator of f can also be interpreted as a kind of gradient step for the function f. In particular, we have

$$\mathbf{prox}_{\lambda f}(v) \approx v - \lambda \nabla f(v) \tag{3}$$

when λ is small and f is differentiable.

- There is a close connection between proximal operators and gradient methods
- The proximal operator may be useful in optimization.
- λ will play a role similar to a step size in a gradient method.

Proximal gradient method

Consider the unconstrained problem with cost function split in two components

$$\min \ f(x) + g(x) \tag{4}$$

- $f : \boldsymbol{R}^n \to \boldsymbol{R}$ convex and differentiable.
- $g : \boldsymbol{R}^n \to \boldsymbol{R} \cup \{+\infty\}$ closed, convex, possibly nondifferentiable;

In this form, we split the objective into two terms, one of which is differentiable.

Proximal gradient method

The proximal gradient method is

$$x^{(k+1)} := \mathbf{prox}_{\lambda^{(k)} g}\left(x^{(k)} - \lambda^{(k)} \nabla f\left(x^{(k)}\right)\right) \tag{5}$$

where $\lambda^{(k)} > 0$ is a step size,constant or determined by line search.

- log-likelihood term $-l(\boldsymbol{\beta_0}, \boldsymbol{\gamma})$ is convex and differentiable which can be regard as f in (4).
- penalized term $\zeta_1 \boldsymbol{J_1}(\boldsymbol{\beta_0}) + \zeta_2 \boldsymbol{J_2}(\boldsymbol{\gamma})$ can be regard as g in (4).
- both the overall penalty term $\boldsymbol{J}_{\zeta_1,\zeta_2}$ and the L_2^2-term can be decomposed into nonoverlapping parts that only contain either $\boldsymbol{\beta_0}$ or $\boldsymbol{\gamma}$.

Proximal gradient method

For $k = 0, 1, 2, \ldots$ until convergence, the proximal gradient iterations in the problem are given by

$$\hat{\beta}_0^{(k+1)} = \mathbf{Prox}_{\zeta_1 / v_{(k)} J_1} \left(\mathbf{v}^{(k)} := \hat{\beta}_0^{(k)} + \frac{1}{v^{(k)}} \cdot \frac{\partial l(\hat{\beta}_0^{(k)}, \hat{\gamma}^{(k)})}{\partial \beta_0} \right) \quad (6)$$

and

$$\hat{\gamma}^{(k+1)} = \mathbf{Prox}_{\zeta_2 / v_{(k)} J_2} \left(\mathbf{w}^{(k)} := \hat{\gamma}^{(k)} + \frac{1}{v^{(k)}} \cdot \frac{\partial l(\hat{\beta}_0^{(k)}, \hat{\gamma}^{(k)})}{\partial \gamma} \right) \quad (7)$$

where $v^{(k)} > 0$ is an inverse stepsize parameter.

The search points $\boldsymbol{v}$ and $\boldsymbol{w}$ for β_0 and γ, respectively, are obtained from a first order approximation of the log-likelihood term in (10) and can be considered a one-step approximation of the ML estimator, based on the current solution.

Analytical solution for γ

Let $J_2(\gamma) = \sum_{j=1}^{p} \phi_j \|\gamma_j\| = \sum_{j=1}^{p} J_{2j}$and let $\boldsymbol{w}$ be partitioned like γ.Then, we can get the analytical solution

$$\mathbf{Prox}_{\zeta_2/v \cdot J_{2j}}(\boldsymbol{w}_{\cdot j}) = \left(1 - \frac{\zeta_2 \phi_j / v}{\|\boldsymbol{w}_{\cdot j}\|}\right)_+ \boldsymbol{w}_{\cdot j}, \quad j = 1, \ldots, p. \tag{8}$$

where $(u)_+ = \max(u, 0)$.

Analytical solution for β_0

Rewrite the penalty on the baseline parameters:

$$\mathbf{J}_1(\boldsymbol{\beta}_0) = \sum_{r=1}^{m}\sum_{t=2}^{q}(\beta_{0tr} - \beta_{0,t-1,r})^2 = \sum_{r=1}^{m}\mathbf{J}_{1r}$$

Let D denote the first-order difference matrix, that is,

$$D = \begin{pmatrix} -1 & 1 & & 0 \\ & -1 & 1 & \\ & & \ddots & \\ 0 & & -1 & 1 \end{pmatrix}$$

With $J_{1r} = \|\boldsymbol{D\beta_{0r}}\|_2^2$ and $\boldsymbol{\Omega} = \boldsymbol{D}^T\boldsymbol{D}$,the analytical solution is :

$$\mathbf{Prox}_{\zeta_1/v\cdot J_{1r}}(\boldsymbol{v}_{\cdot r}) = (\boldsymbol{I} + \frac{\zeta_1}{v}\boldsymbol{\Omega})^{-1}\boldsymbol{v}_{\cdot r}, \quad r = 1, \ldots, m \qquad (9)$$

Tuning parameter selection

The tuning parameters ζ_1 and ζ_2 are chosen by k-fold CV(cross-validation),but needs a modification to standard.

For folds $s = 1, \ldots, k$,we let

- $\mathbf{I}_s$:the index set of observations that belong to fold s.
- $\hat{\lambda}_r^{(-s)}$:the estimate for λ_r that is based on all observations except for those in $\mathbf{I}_s$

Tuning parameter selection

We use the (predictive) deviance as the criterion to be cross-validated.

The cross-validated deviance is defined by

$$D_{CV} = 2 \sum_{s=1}^{K} \sum_{i \in I_s} \sum_{t=1}^{t_i} \sum_{r=0}^{m} y_{itr} log \left(\frac{y_{itr}}{\hat{\lambda}_r(t|\mathbf{x}_i)^{(-s)}} \right) \tag{10}$$

In which we can see that all $(m+1) \cdot t_i$ data points y_{itr} belong to the same original observation i assigned to the same cross-validation fold.