Chapter 2, Optimization

Newton's method with a large p

Ying Wei & Xiaoqi Lu

February 26, 2020

1 Newton's method with a large p

- Recall the optimization $\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta} \in R^p} f(\boldsymbol{\theta})$
- Newton's method suggests to update $\boldsymbol{\theta}$ iteratively, such that the ith step is given by

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} - \left[\nabla^2 f(\boldsymbol{\theta}_{i-1})\right]^{-1} \nabla f(\boldsymbol{\theta}_{i-1}),$$

  where $\nabla f(\boldsymbol{\theta}_{i-1})$ is the gradient, and $\nabla^2 f(\boldsymbol{\theta}_{i-1})$ is the Hessian matrix.

- Question: is Newtown's method scalable with increasing number of parameters p?
  - The computational burden in calculating the inverse of the Hessian Matrix $\left[\nabla^2 f(\boldsymbol{\theta}_{i-1})\right]^{-1}$ increases quickly with p.

1.1 Quasi-Newton Methods

Ascent direction: For a function f, a direction $\mathbf{d}$ is an ascent direction for f at a given point $\boldsymbol{\theta}_0$ if there exists some $\epsilon > 0$ such that

$$f(\boldsymbol{\theta}_0 + \lambda \mathbf{d}) > f(\boldsymbol{\theta}_0)$$

for all $0 < \lambda < \epsilon$.

Directional derivative: The derivative of a function $f :^p \to$ at $\boldsymbol{\theta}$ in the direction of $\mathbf{d}$ is defined by

$$\lim_{\lambda \to 0} \frac{f(\boldsymbol{\theta} + \lambda \mathbf{d}) - f(\boldsymbol{\theta})}{\lambda} = \frac{\partial}{\partial \lambda} f(\boldsymbol{\theta} + \lambda \mathbf{d}) \bigg|_{\lambda=0} = \mathbf{d}' \nabla f(\boldsymbol{\theta}).$$

From this definition, we can see that $\mathbf{d}$ is an ascent direction for f at $\boldsymbol{\theta}_0$ if and only if $\mathbf{d}' \nabla f(\boldsymbol{\theta}_0) > 0$.

Newton's method updates the parameters by

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \left[\nabla^2 f(\boldsymbol{\theta}_i)\right]^{-1} \nabla f(\boldsymbol{\theta}_i),$$

Newton's direction $\mathbf{d} = -\left[\nabla^2 f(\boldsymbol{\theta}_i)\right]^{-1} \nabla f(\boldsymbol{\theta}_i)$ is an ascent direction if $\left[\nabla^2 f(\boldsymbol{\theta}_i)\right]^{-1}$ is negative definite.

More general: One can update

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \mathbf{H}_{i,p\times p} \nabla f(\boldsymbol{\theta}_i),$$

and $f(\boldsymbol{\theta}_{i+1}) > f(\boldsymbol{\theta}_i)$ for any $\mathbf{H}_{i,p\times p}$ that is positive definite.

- Gradient Descent Algorithm: $\mathbf{H}_i = I_{p\times p}$ for any i
- Easy to compute, but could slow in convergence.

1.1.1 Other Quasi-Newton Iterations

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \lambda_i [\mathbf{B}_i]^{-1} \nabla f(\boldsymbol{\theta}_i)$$

Question: can we find a surrogate matrix $\mathbf{B}_i$ that is similar to $\nabla^2 f(\boldsymbol{\theta}_i)$, but easier to compute?

- Consider Taylor expansion on $\nabla f(\boldsymbol{\theta}_{i-1})$ around $\boldsymbol{\theta}_i$:

$$\nabla f(\boldsymbol{\theta}_i) - \nabla f(\boldsymbol{\theta}_{i-1}) = \nabla^2 f(\boldsymbol{\theta}_i) \cdot (\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}) + o\left(\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}\|\right) \quad (1)$$

- A good approximation can be achieved $\mathbf{B}_i$ satisfies the following **secant equation**

$$\mathbf{B}_i \mathbf{S}_{i-1} = \mathbf{Y}_{i-1} \tag{2}$$

where

$$\mathbf{Y}_{i-1} = \nabla f(\boldsymbol{\theta}_i) - \nabla f(\boldsymbol{\theta}_{i-1})$$

$$\mathbf{S}_{i-1} = (\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1})\,.$$

- There are infinitely many $\mathbf{B}_i$ satisfying the **secant equation**.

SR1 (Symmetric-Rank-1) Method

Assuming a simple structural constrain that

$$\mathbf{B}_i = \mathbf{B}_{i-1} + \sigma v v^T,$$

where v is a p -dimensional vector. Combined with the secant equation, we have

$$\begin{aligned}
&\mathbf{Y}_{i-1} = \mathbf{B}_{i-1}\mathbf{S}_{i-1} + \sigma v^T \mathbf{S}_{i-1} v \\
\Rightarrow & v = \delta(\mathbf{Y}_{i-1} - \mathbf{B}_{i-1}\mathbf{S}_{i-1}) \quad \text{for some } \delta \in \mathbb{R} \\
\Rightarrow & (\mathbf{Y}_{i-1} - \mathbf{B}_{i-1}\mathbf{S}_{i-1}) = \sigma\delta^2 \left[\mathbf{S}_{i-1}^T(\mathbf{Y}_{i-1} - \mathbf{B}_{i-1}\mathbf{S}_{i-1})\right](\mathbf{Y}_{i-1} - \mathbf{B}_{i-1}\mathbf{S}_{i-1})
\end{aligned}$$

Choose $\sigma = \operatorname{sign}\left[\mathbf{S}_{i-1}^T(\mathbf{Y}_{i-1} - \mathbf{B}_{i-1}\mathbf{S}_{i-1})\right]$ and $\delta = \left|\mathbf{S}_{i-1}^T(\mathbf{Y}_{i-1} - \mathbf{B}_{i-1}\mathbf{S}_{i-1})\right|^{-1/2}$, thus

$$\mathbf{B}_i = \mathbf{B}_{i-1} + \frac{(\mathbf{Y}_{i-1} - \mathbf{B}_{i-1}\mathbf{S}_{i-1})(\mathbf{Y}_{i-1} - \mathbf{B}_{i-1}\mathbf{S}_{i-1})^T}{\mathbf{S}_{i-1}^T(\mathbf{Y}_{i-1} - \mathbf{B}_{i-1}\mathbf{S}_{i-1})} \tag{3}$$

DFP (Davidon-Fletcher-Powell) Method

DFP Update

$$\min_{\mathbf{B}} \quad \|\mathbf{B} - \mathbf{B}_{i-1}\| \tag{4}$$

$$\text{subject to} \quad \mathbf{B} = \mathbf{B}^T,\ \mathbf{B}\mathbf{S}_{i-1} = \mathbf{Y}_{i-1} \tag{5}$$

Solution:

$$\mathbf{B}_i = (I - \frac{\mathbf{Y}_{i-1}\mathbf{S}_{i-1}^T}{\mathbf{Y}_{i-1}^T\mathbf{S}_{i-1}})\mathbf{B}_k(I - \frac{\mathbf{S}_{i-1}\mathbf{Y}_{i-1}^T}{\mathbf{Y}_{i-1}^T\mathbf{S}_{i-1}}) + \frac{\mathbf{Y}_{i-1}\mathbf{Y}_{i-1}^T}{\mathbf{Y}_{i-1}^T\mathbf{S}_{i-1}} \tag{6}$$

BFGS (Broyden-Fletcher-Goldfarb-Shanno) Method*

To avoid taking inverse of $\mathbf{B}$, BFGS propose to approximate $\nabla^2 f(\boldsymbol{\theta}_i)^{-1}$ directly. Similar to DFP, they propose the following optimization problem:

$$\min_{\mathbf{H}} \quad \|\mathbf{H} - \mathbf{H}_{i-1}\| \tag{7}$$

$$\text{subject to} \quad \mathbf{H} = \mathbf{H}^T, \mathbf{H}\mathbf{Y}_{i-1} = \mathbf{S}_{i-1} \tag{8}$$

Solution:

$$\mathbf{H}_i = (I - \frac{\mathbf{S}_{i-1}\mathbf{Y}_{i-1}^T}{\mathbf{Y}_{i-1}^T\mathbf{S}_{i-1}})\mathbf{H}_{i-1}(I - \frac{\mathbf{Y}_{i-1}\mathbf{S}_{i-1}^T}{\mathbf{Y}_{i-1}^T\mathbf{S}_{i-1}}) + \frac{\mathbf{S}_{i-1}\mathbf{S}_{i-1}^T}{\mathbf{Y}_{i-1}^T\mathbf{S}_{i-1}} \tag{9}$$

Note: BFGS is more effective than most quasi-Newton methods, and is the "go-to" method in many optimization problems.

1.2 Coordinate-wise optimization

Another simple approach is to consider coordinate descent approach, that starts with initial guess of $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \cdots, \theta_p^{(0)})$, and then update one component of $\boldsymbol{\theta}^{(0)}$ at a time iteratively

$$
\begin{aligned}
\theta_1^{(i+1)} &= \arg\max_{\theta_1} f(\theta_1, \theta_2^{(i)}, \cdots, \theta_p^{(i)}) \\
\theta_2^{(i+1)} &= \arg\max_{\theta_2} f(\theta_1^{(i+1)}, \theta_2, \theta_3^{(i)} \cdots, \theta_p^{(0)}) \\
\theta_3^{(i+1)} &= \arg\max_{\theta_3} f(\theta_1^{(i+1)}, \theta_2^{(i+1)}, \theta_3, \theta_4^{(0)} \cdots, \theta_p^{(0)}) \\
\vdots &= \vdots \\
\theta_p^{(i+1)} &= \arg\max_{\theta_p} f(\theta_1^{(i+1)}, \theta_2^{(i+1)}, \cdots, \theta_{p-1}^{(i+1)}, \theta_p)
\end{aligned}
$$

Question: are we able to reach the global optimizer by maximizing / minmizing along each coordinate axis?

Answers: Yes, if $f(\boldsymbol{\theta})$ is convex and differentiable; $f(\boldsymbol{\theta})$ is convex but not differentiable; the coordinate descent approach still works if there exist a convex and differentiable $g()$ and convex $h_k(\theta_k)$ for each $k = 1, ..., p$ such that $f(\boldsymbol{\theta}) = g(\boldsymbol{\theta}) + \sum_{k=1}^{p} h_k(x_k)$.

- Solving p one-dimensional optimization is easier than solving one p-dimensional optimization.
- Order of cycle through coordinates is arbitrary
- One can replace individual coordinates with blocks of coordinates

Exercise 1.1 *Consider a linear regression $EY = \mathbf{X}^T\boldsymbol{\beta}$, where Y is response vector, $\boldsymbol{\beta}$ is p-dimensional coefficient, and $\mathbf{X}$ is the design matrix with columns $\mathbf{X}_1, \cdots, \mathbf{X}_p$. The LS loss function*
$f(\beta) = \|Y - \mathbf{X}^T\boldsymbol{\beta}\|^2$

Consider minimizing over β_k while fixing all $\beta_j, j \neq k$

$$0 = \nabla_k f(\boldsymbol{\beta}) = \mathbf{X}_k^T(\mathbf{X}^T\boldsymbol{\beta} - Y) = \mathbf{X}_k^T(\mathbf{X}_k\beta_k + \mathbf{X}_{-k}^T\boldsymbol{\beta}_{-k} - Y)$$

$$\Rightarrow \beta_k = \frac{\mathbf{X}_k^T(Y - \mathbf{X}_{-k}^T\boldsymbol{\beta}_{-k})}{\mathbf{X}_k^T\mathbf{X}_k}$$

Coordinate descent repeats this update for k = 1,2,...,p,1,2,...

1.3 Regularized regressions for high-dimensional p

Regularization is the common variable selection approaches for high-dimensional covariates. The best known Regularization is called LASSO. In linear regression, LASSO minimize

$$f(\boldsymbol{\beta}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{i,j}\beta_j)^2 + \gamma\sum_{j=1}^{p}|\beta_j|$$

for some $\gamma \geq 0$. Here the $x_{i,j}$ are standardized so that $\sum_i x_{i,j}/n = 0$ and $\sum_i x_{i,j}^2 = 1$.

With a single predictor x, the lasso solution is very simple

$$\widehat{\beta}^{lasso}(\gamma) = S(\widehat{\beta}, \gamma) = sign(\widehat{\beta})(|\widehat{\beta}| - \gamma)_+$$

$$\widehat{\beta}^{\text{lasso}}(\gamma) = S(\widehat{\beta}, \gamma) = \begin{cases} \widehat{\beta} - \gamma, \ \text{if } \widehat{\beta} > 0 \text{ and } \gamma < |\widehat{\beta}| \\ \widehat{\beta} + \gamma, \ \text{if } \widehat{\beta} < 0 \text{ and } \gamma < |\widehat{\beta}| \\ 0, \ \text{if } \gamma > |\widehat{\beta}| \end{cases}$$

- $S(\widehat{\beta}, \gamma)$ is called soft threshold.
- If x are not standardized, i.e. $\langle x, x \rangle = \sum_i x_i^2 \neq 1$

$$\widehat{\beta}^{\text{lasso}}(\gamma) = \frac{S(\langle x, y \rangle, \gamma)}{\langle x, x \rangle} = S(\widehat{\beta}, \frac{\gamma}{\langle x, x \rangle})$$

- If multiple predictors that are uncorrelated/orthogonal (i.e. $\langle X_i, X_j \rangle = 0$), the lasso solutions are soft-thresholded versions of the individual least squares estimates.
- That is not the case when predictors are correlated. When p is large, the optimization could be challenging.

A coordinate-wise descent algorithm

- Coordinate-wise objective function

$$f(\beta_j) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \sum_{k\neq j} x_{i,k}\widetilde{\beta}_k - x_{i,j}\beta_j)^2 + \gamma\sum_{k\neq j}|\widetilde{\beta}_k| + \gamma|\beta_j|$$

- Minimizing $f(\beta_j)$ w.r.t. β_j while having $\widetilde{\beta}_k$ fixed, we have

$$\widetilde{\beta}_j(\gamma) \leftarrow S\left(\sum_{i=1}^{n} x_{i,j}(y_i - \tilde{y}_i^{(-j)}), \gamma\right) \tag{10}$$

  where $\tilde{y}_i^{(-j)} = \sum_{k\neq j} x_{i,k}\widetilde{\beta}_k$. That is equivalent to regressing the partial residual $y_i - \tilde{y}_i^{(-j)}$ against $x_{i,j}$. The soft-threshold holds.

- We can then update β_j repeatedly for j = 1,2,...,p,1,2,... until convergence.

Covariance Updates and its flexibility with sparse matrix Note that $y_i - \tilde{y}_i^{(-j)} = y_i - \widehat{y}_i + x_{i,j}\widetilde{\beta}_j = r_i + x_{i,j}\widetilde{\beta}_j$, where $\widehat{y}_i$ is fitted value at "current parameter" and r_i is the current residual.

$$
\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} x_{i,j}(y_i - \tilde{y}_i^{(-j)}) &= \frac{1}{n}\sum_{i=1}^{n} x_{i,j} r_i + \widetilde{\beta}_j \\
&= \frac{1}{n}\left\{ \langle x_j, y\rangle - \sum_{k:|\widetilde{\beta}_k|>0} \langle x_j, x_k\rangle \widetilde{\beta}_k \right\} + \widetilde{\beta}_j
\end{aligned}
$$

where the inner product $\langle x_j, y\rangle = \sum_{i=1}^{n} x_{i,j}, y_i$.

Sparse coding is an efficient way to store large sparse matrix, where we store only the non-zero entries and the coordinates where they occur.

Weighted Updates

- Often a weight w_i is associated with each observation.
- In this case, the regression coefficients is equivalent to regress $\sqrt{w_i}y_i$ against $\sqrt{w_i}x_i$

$$\sum_i w_i(y_i - x_i^\top \beta)^2 \Leftrightarrow \sum_i(\sqrt{w_i}y_i - \sqrt{w_i}x_i^\top \beta)^2$$

- The lasso update becomes only slightly more complicated:

$$\widetilde{\beta}_j(\gamma) \leftarrow \frac{S\left(\sum_i w_i x_{i,j}(y_i - \tilde{y}_i^{(-j)}), \gamma\right)}{\sum_i w_i x_{i,j}^2} \tag{11}$$

Pathwise coordinatewise optimization algorithms

1. Starting at the smallest value λ for which the entire vector $\widehat{\beta} = 0$.
   $\lambda_{\text{max}} = \max_l \langle X_l, y \rangle$
2. Compute the solution sequentially at sequence
   $\lambda_{\text{max}} \geq \lambda_1 \geq \cdots \geq \lambda_{\text{min}} \geq 0$
3. For tuning parameter value λ_{k+1}, initialize coordinate descent algorithm at the computed solution for λ_k (warm start)

1.3.1 Extension of CD to logistic regression

Recall that the log likelihood of a logistic regression

$$f(\beta_0, \boldsymbol{\beta}_1) = \sum_{i=1}^{n} \left(y_i(\beta_0 + \boldsymbol{\beta}_1^T \mathbf{x}_i) - \log\left(1 + e^{\beta_0 + \boldsymbol{\beta}_1^T \mathbf{x}_i}\right)\right).$$

The gradient of this function is

$$\nabla f(\beta_0, \beta_1) = \begin{pmatrix} \sum_{i=1}^n y_i - p_i \\ \sum_{i=1}^n \mathbf{x}_i(y_i - p_i) \end{pmatrix}_{(p+1)\times 1}, \tag{12}$$

where $p_i = P(Y_i = 1|\mathbf{x}_i) = \frac{exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}_1)}{1+exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}_1)}$.

The Hessian is given by

$$\begin{aligned} \nabla^2 f(\beta_0, \boldsymbol{\beta}_1) &= -\sum_{i=1}^n \begin{pmatrix} 1 \\ \mathbf{x}_i \end{pmatrix} \begin{pmatrix} 1 & \mathbf{x}_i^T \end{pmatrix} p_i(1-p_i) \\ &= -\begin{pmatrix} \sum p_i(1-p_i) & \sum \mathbf{x}_i^T p_i(1-p_i) \\ \sum \mathbf{x}_i p_i(1-p_i) & \sum \mathbf{x}_i \mathbf{x}_i^T p_i(1-p_i) \end{pmatrix}. \end{aligned}$$

A quadratic approximation to the log-likelihood $f(\beta_0, \boldsymbol{\beta}_1)$

If we Taylor expansion the log-likelihood around "current estimates" $(\widetilde{\beta}_0, \widetilde{\boldsymbol{\beta}}_1)$, we have

$$f(\beta_0, \boldsymbol{\beta}_1) \approx \ell(\beta_0, \boldsymbol{\beta}_1) = -\frac{1}{2n} \sum_{i=1}^{n} w_i (z_i - \beta_0 - \mathbf{x}_i^T \boldsymbol{\beta}_1)^2 + C(\widetilde{\beta}_0, \widetilde{\boldsymbol{\beta}}_1)$$

where

$$z_i = \widetilde{\beta}_0 + \mathbf{x}_i^T \widetilde{\boldsymbol{\beta}}_1 + \frac{y_i - \widetilde{p}_i(\mathbf{x}_i)}{\widetilde{p}_i(\mathbf{x}_i)(1 - \widetilde{p}_i(\mathbf{x}_i))} \text{ working response}$$

$$w_i = \widetilde{p}_i(\mathbf{x}_i)(1 - \widetilde{p}_i(\mathbf{x}_i)), \text{ working weights}$$

$$\widetilde{p}_i = \frac{\exp(\widetilde{\beta}_0 + \mathbf{x}_i^T \widetilde{\boldsymbol{\beta}})}{1 + \exp(\widetilde{\beta}_0 + \mathbf{x}_i^T \widetilde{\boldsymbol{\beta}}_1)} \text{(probability evaluated at the current parameters)}$$

Exercise 1.2 *The logistic-lasso can be written as a penalized weighted least-squares problem*

$$\min_{(\beta_0,\boldsymbol{\beta}_1)} L(\beta_0,\boldsymbol{\beta}_1,\lambda) = \{-\ell(\beta_0,\boldsymbol{\beta}_1) + \lambda \sum_{j=0}^{p} |\beta_j|\}$$

Derive path-wise coordinate descendent algorithm update for the optimization above

Step 1 *Find* λ_{max} *such that all the estimated* β *are zero;*

Step 2 *Define a fine sequence* $\lambda_{max} \geq \lambda_1 \geq \cdots \geq \lambda_{min} \geq 0$

Step 3 *Defined the quadratic approximated objective function* $L(\beta_0,\boldsymbol{\beta}_1,\lambda)$ *for* λ_k *using the estimated parameter at* λ_{k-1} *(*$\lambda_{k-1} > \lambda_k$*).*

Step 4 *Run coordinate descendent algorithm to find the optimization defined in Step 3*