Coordinate descent 法と Lasso への応用について

B2 武山尚生

2020 年 3 月 3 日

1 Coordinate descent 法

Coordinate descent 法とは、p 個あるパラメータのうちの p-1 個を固定し、1 つの変数に関してのパラメータの更新を行う。これを全ての変数で反復し、目的関数を最小化する手法である。利点として、一回あたりの反復にかかる計算時間・メモリが他の手法より小さいこと、極めて多くのパラメータを持つ関数でも、計算量が増大しにくいことが挙げられる。欠点としては収束が遅いことが挙げられる。

2 Lasso への応用

Lasso は、通常の重回帰分析に加え、L1 ノルムで正則化したものである。回帰係数が 0 に収束しやすく、重回帰分析よりスパースな推定が可能である。Lasso では以下の目的関数を最小化するパラメータを推定する。

$$\begin{aligned} L_a(\beta) &= \frac{1}{2n}|y - X\beta|^2 + a|\beta| \\ &= \frac{1}{2n}\sum_{i=1}^{N}(y_i - \sum_{j=1}^{p}\beta_j x_{ij})^2 + a\sum_{j=1}^{p}|\beta_j| \end{aligned} \tag{1}$$

$a|\beta|$ は $\beta_j = 0$ の時微分できないため、以下の劣勾配で表す。

$$d_j \in \begin{cases} -1, & (\beta_j < 0) \\ [-1, 1], & (\beta_j = 0) \\ 1, & (\beta_j > 0) \end{cases}$$

このように、L1 ノルムの微分係数を場合分けで表す。これを用いて目的関数の微分を行うと、以下の式が得られる。

$$\begin{aligned} \frac{\partial}{\partial\beta_j}L_a(\beta) &= -\frac{1}{n}\sum_{i=1}^{N}x_{ij}(y_i - \sum_{k=1}^{p}\beta_k x_{ik}) + ad_j = 0 \\ &\Leftrightarrow \frac{1}{n}\sum_{i=1}^{N}x_{ij}(y_i - \sum_{k\neq j}^{p}\beta_k x_{ik} - \beta_j x_{ij}) = ad_j \\ &\Leftrightarrow \beta_j\frac{1}{n}\sum_{i=1}^{N}x_{ij}^2 = \frac{1}{n}\sum_{i=1}^{N}x_{ij}(y_i - \sum_{k\neq j}^{p}\beta_k x_{ik}) - ad_j \\ &\Leftrightarrow \beta_j = \frac{1}{n}\sum_{i=1}^{N}x_{ij}(y_i - \sum_{k\neq j}^{p}\beta_k x_{ik}) - ad_j \end{aligned} \tag{2}$$

よって、$beta_j$ に関する推定方程式を得ることが出来る。ここで、(2) における第 j 変数に対応する項を除いた残差ベクトルを $r^(j)$ で表す。

$$r^{(j)} = y - \sum_{k \neq j} x_{(k)} \tilde{\beta_k}$$

すると、(2) 式は以下のように変形できる。

$$\beta_j = \frac{1}{n} \sum_{i=1}^{N} x_{ij} r_i^{(j)} - ad_j \tag{3}$$

ここで、Soft thresholding function を以下のように定義する。

$$S(x, \lambda) = sign(x)(|x| - \lambda)_+$$

すると (2) 式は以下のように表せる。

$$\beta_j = S(\frac{1}{n} r^{(j)T} x_{(j)}, a) \tag{4}$$

よって、ベクトルの要素である β_j の更新式が得られた。この更新式は、Coordinate descent 法では、β が収束するまでパラメータの更新を反復し、差分が一定値を下回ったら更新を打ち切る。以上が Coordinate descent 法の導出である。
また、ここで用いた Proximal Gradient Method の利点としては、関数 $L(x) = f(x) + g(x)$ のうち $g(x)$ に微分出来ない点があった場合でも、以下のように更新式を定めることで目的関数を最小化出来る点にある。

$$x_{k+1} = prox_\eta g(x_k - \eta f(x_k))$$
$$prox_g(y) = argmin_x \{g(x) + \frac{1}{2}|x - y|^2\}$$

近接写像 $prox_g(y)$ の定義について詳しく説明するため、更に式を展開する。

$$\begin{aligned}
x_{k+1} &= \mathrm{prox}_{\eta g}(x_k - \eta f(x_k)) \\
&= \mathrm{argmin}_x \left(\eta g(x) + \frac{1}{2} \parallel x - (x_k - \eta f(x_k)) \parallel^2 \right) \\
&= \mathrm{argmin}_x \left(g(x) + f(x_k) + \frac{1}{2\eta} \parallel x - x_k + \eta f(x_k)) \parallel^2 \right) \\
&= \mathrm{argmin}_x \left(g(x) + f(x_k) + f(x_k)^T (x - x_k) + \frac{1}{2\eta} \parallel x - x_k \parallel^2 \right) \\
&= \mathrm{argmin}_x \left(g(x) + \hat{f}_\eta(x; x_k) \right) \\
\hat{f}_\eta(x; y) &= f(y) + f(y)^T (x - y) + \frac{1}{2\eta} \parallel x - y \parallel^2
\end{aligned} \tag{5}$$

適切な η をとれば、近接写像 prox(x) は L(x) の y における二次近似となっている。(1) を近接写像 prox と見た時、β に関して偏微分することで最小値を求めている。これは L(x) における最小値を、y を中心とした f(x) の二次近似である (5) の最小値を逐次更新することで求めていることと同じである。よって、Proximal Gradient Method は局所的に微分不可能な目的関数でも最適化出来る。

3 Group Lasso への応用

Group Lasso は通常の重回帰分析に加えて、説明変数が属するグループ毎に L2 ノルムを正則化項としたものである。通常の重回帰分析に対して、グループ毎に回帰係数が全て 0 になる推定ができ、説明変数に用いるグループを絞ったスパースな推定が可能である。Group Lasso では以下の目的関数を最小化する。

$$F(x) = f(x) + \lambda \sum_{j=1}^{J} |x_{G_j}|^2 \tag{6}$$

但し、x_{g_j} は j 番目のグループに属する説明変数の集合を表している。
これを Proximal Gradient Method の定義に従って二次近似すると、以下の更新式が得られる。

$$h(x) = f(x_k)^T (x - x_k) + \frac{1}{2\eta}||x - x_k||^2 + g(x) \tag{7}$$

$$\frac{\partial h(x)}{\partial x} = \nabla f(x_k) + \frac{1}{\eta}(x - x_k) + \lambda \frac{\partial}{\partial x} \sum_{j=1}^{J} ||x_{G_j}|| \tag{8}$$

正則化項はそのまま x では微分できないため、要素 x_i が j に属していると考える。この時、(8) は以下のように変形できる。

$$\frac{\partial h(x)}{\partial x_i} = \nabla f(x_k)_i + \frac{1}{\eta}(x_i - x_{ki}) + \lambda \frac{\partial}{\partial x_i} ||x_{G_j}|| \tag{9}$$

$$= \nabla f(x_k)_i + \frac{1}{\eta}(x_i - x_{ki}) + \lambda \frac{x_i}{||x_{G_j}||} \tag{10}$$

これを $\frac{\partial h(x)}{\partial x_i} = 0$ について解くと、

$$x_i = x_{ki} - \eta \nabla f(x_k)_i - \eta\lambda \frac{x_i}{||x_{G_j}||} \tag{11}$$

ここで、$\tilde{x}_k := x_k - \eta \nabla f(x_k)$ とすると、以下の式が得られる。

$$x_i = \tilde{x}_{ki} - \eta\lambda \frac{x_i}{||x_{G_j}||} \tag{12}$$

グループ毎に同じ更新式が適用できるため、以下のように書き換えられる。

$$x_{Gj} = \tilde{x}_{k,G_j} - \eta\lambda \frac{x_{Gj}}{||x_{G_j}||} \tag{13}$$

ここで、以下のように α を置く。

$$x_{Gj} = \alpha \tilde{x}_{k,G_j} \tag{14}$$

これを (13) に代入すると、

$$\alpha \left(1 + \frac{\eta\lambda}{|\alpha|\,||\tilde{x}_{k,G_j}||}\right) = 1 \tag{15}$$

$$\Leftrightarrow \alpha \left(|\alpha|\,||\tilde{x}_{k,G_j}|| + \eta\lambda\right) = |\alpha|\,||\tilde{x}_{k,G_j}|| \tag{16}$$

λ, η が共に正であることから、α も正である。よって、α は以下のように表せる。

$$\alpha = 1 - \frac{\eta\lambda}{||\tilde{x}_{G_j}||} \tag{17}$$

α が正であることから、

$$\alpha = \max\left(0, 1 - \frac{\eta\lambda}{||\tilde{x}_{G_j}||}\right) \tag{18}$$

これを (14) に代入することで、x_{G_j} に関する更新式となる。

$$x_{G_j} = \max\left(0, 1 - \frac{\eta\lambda}{||\tilde{x}_{G_j}||}\right) \tilde{x}_{k,G_j} \tag{19}$$

得られた x_{G_j} は (7) の h(x) を最小化することから、各変数に関して h(x) を更新して、(6) の F(x) の値を最小化できる。また、更新式は x の各変数に関して得られているため、変数一つを取り出して最適化することが可能である。よって、Coordinate descent 法が適用でき、単一の変数に対しての最適化を繰り返すことで、F(x) を最小化させる x を推定できる。以上が Group Lasso における Coordinate descent 法の導出である。

参考文献

「python ではじめる Group Lasso」 https://qiita.com/AnchorBlues/items/4e50d3b98a40c8b3086e
「近接勾配法 (Proximal Gradient Method)」 https://qiita.com/msekino/items/9f217fcd735513627f65
「座標降下法による重回帰分析 with lasso の係数推定プログラム」https://qiita.com/m1t0/items/9af55f937a742a8d7f2a