Python中优化方法

statsmodels.discrete.discrete_model.Logit.fit(*start_params=None*, *method='newton'*, *maxiter=35*, *full_output=1*, *disp=1*, *callback=None*, ***kwargs*)

method可选择的有：

- 'newton'：Newton-Raphson
- 'nm'：Nelder-Mead Simplex algorithm
- 'bfgs'：Broyden-Fletcher-Goldfarb-Shanno algorithm
- 'lbfgs'：limited-memory BFGS with optional box constraints
- 'powell'：modified Powell’s method
- 'cg'：conjugate gradient
- 'ncg'：Newton-Conjugate-Gradient algorithm
- 'basinhopping'：global basin-hopping solver
- 'minimize'：generic wrapper of scipy minimize (BFGS by default)

sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None)

solver{‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’}, default=’lbfgs’

- For small datasets, ‘liblinear’ is a good choice, whereas ‘sag’ and ‘saga’ are faster for large ones.

- For multiclass problems, only ‘newton-cg’, ‘sag’, ‘saga’ and ‘lbfgs’ handle multinomial loss; ‘liblinear’ is limited to one-versus-rest schemes.
- ‘newton-cg’, ‘lbfgs’, ‘sag’ and ‘saga’ handle L2 or no penalty
- ‘liblinear’ and ‘saga’ also handle L1 penalty
- ‘saga’ also supports ‘elasticnet’ penalty
- ‘liblinear’ does not support setting `penalty='none'`

Note that ‘sag’ and ‘saga’ fast convergence is only guaranteed on features with approximately the same scale. You can preprocess the data with a scaler from sklearn.preprocessing.

New in version 0.17: Stochastic Average Gradient descent solver.

New in version 0.19: SAGA solver.

优化方法

符号说明：

- $g_k = g(x^{(k)}) = \nabla f(x^{(k)})$为$f(x)$在$x^{(k)}$处的梯度
- λ_k为步长，$\lambda_k \geq 0$
- $H_k = H(x^{(k)}) = [\frac{\partial^2 f}{\partial x_i^{(k)} \partial x_j^{(k)}}]_{n\times n}$为Hessian矩阵
- G_k近似H_k^{-1}，B_k近似H_k
- $y_k = g_{k+1} - g_k$，$\delta_k = x^{(k+1)} - x^{(k)}$
- 牛顿条件：$\delta_k = H_k^{-1} y_k$或$y_k = H_k \delta_k$
-

不同算法：

- 梯度下降gradient descent（又称“最速下降法steepest descent”）

$$\min_x f(x) \qquad f(x)\text{一阶可导} \tag{1}$$
$$f(x) = f(x^{(k)}) + g_k^T (x - x^{(k)}) \tag{2}$$
$$\Rightarrow x^{(k+1)} = x^{(k)} - \lambda_k g_k \tag{3}$$

- Newton method

$$\min_x f(x) \qquad f(x)\text{二阶可导} \tag{4}$$
$$f(x) = f(x^{(k)}) + g_k^T(x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T H_k (x - x^{(k)}) \tag{5}$$
$$\Rightarrow \nabla f(x) = g_k + H_k(x - x^{(k)}) \tag{6}$$
$$\Rightarrow x^{(k+1)} = x^{(k)} - H_k^{-1} g_k \tag{7}$$

$g_{k+1} - g_k = H_k(x^{(k+1)} - x^{(k)}) \Rightarrow y_k = H_k \delta_k$

quasi-Newton method将Newton method中的H_k^{-1}用其他矩阵来逼近

- quasi-Newton method: DFP algotrithm (Davidon-Fletcher-Powell)

$$G_{k+1} = G_k + P_k + Q_k \tag{8}$$
$$G_{k+1} y_k = G_k y_k + P_k y_k + Q_k y_k \tag{9}$$
$$\text{令} P_k y_k = \delta_k, Q_k y_k = -G_k y_k \tag{10}$$
$$\Rightarrow P_k = \frac{\delta_k \delta_k^T}{\delta_k^T y_k}, Q_k = -\frac{G_k y_k y_k^T G_k}{y_k^T G_k y_k} \tag{11}$$
$$\Rightarrow G_{k+1} = G_k + \frac{\delta_k \delta_k^T}{\delta_k^T y_k} - \frac{G_k y_k y_k^T G_k}{y_k^T G_k y_k} \tag{12}$$
$$\Rightarrow x^{(k+1)} = x^{(k)} - \lambda_k G_k g_k \tag{13}$$

初始G_0须设定为正定矩阵，则迭代过程中的每个G_k都是正定的.

- quasi-Newton method: BFGS algotrithm (Broyden-Fletcher-Goldfarb-Shannon)

$$B_{k+1} = B_k + P_k + Q_k \tag{14}$$
$$B_{k+1}\delta_k = B_k \delta_k + P_k \delta_k + Q_k \delta_k \tag{15}$$
$$\text{令} P_k \delta_k = y_k, Q_k \delta_k = -B_k \delta_k \tag{16}$$
$$\Rightarrow B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T \delta_k} - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T \delta_k \delta_k} \tag{17}$$
$$G_k = B_k^{-1} \Rightarrow G_{k+1} = (I - \frac{\delta_k y_k^T}{\delta_k^T y_k}) G_k (I - \frac{\delta_k y_k^T}{\delta_k^T y_k})^T + \frac{\delta_k \delta_k^T}{\delta_k^T y_k} \tag{18}$$
$$\Rightarrow x^{(k+1)} = x^{(k)} - \lambda_k B_k^{-1} g_k, \quad x^{(k+1)} = x^{(k)} - \lambda_k G_k g_k \tag{19}$$

初始G_0须设定为正定矩阵，则迭代过程中的每个G_k都是正定的.

- quasi-Newton method: Broyden's algotrithm

$$G_{k+1} = \alpha G^{DFP} + (1-\alpha) G^{BFGS}, 0 \leq \alpha \leq 1$$

其中G^{DFP}表示DFP算法计算出的G，G^{BFGS}表示BFGS算法计算出的G.

坐标轴下降法 求解Lasso回归

coordinate descent：一个可微的凸函数$J(\theta)$，其中θ是$n \times 1$的向量，即有n个维度。若在某一点$\bar{\theta}_i(i=1,2,\dots,n)$上都是最小值，则$J(\bar{\theta}_i)$就是一个全局的最小值。

1. 初始化$\theta^{(0)}$
2. 从$\theta_1^{(k)}$到$\theta_n^{(k)}$，依次求$\theta_i^{(k)}$，$\theta_i^{(k)} = \arg\min_{\theta_i} J(\theta_1^{(k)}, \theta_2^{(k)}, \dots, \theta_{i-1}^{(k)}, \theta_i$，$\theta_{i+1}^{(k-1)}, \dots, \theta_n^{(k-1)})$
3. 迭代至在所有维度上变化都比较小

最小角回归法 求解Lasso回归

Logistic回归

1. 指数族分布角度的Logistic回归

(1)指数族分布

指数族分布$f(y;\theta,\phi) = exp\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\}$，其中$\theta$为自然参数，$\phi$为散度参数，$a,b,c$为函数，且满足以下条件：

- $a(\phi) > 0$，连续，通常为ϕ/w，其中w为已知先验权重
- $b(\theta)$二阶导数存在且大于零
- $c(y,\phi)$与参数θ无关

指数族分布的性质如下：

$$E(Y) = b^{'}(\theta) \tag{20}$$
$$Var(Y) = b^{''}(\theta)a(\phi) = V(\mu)a(\phi), \text{其中} V(\mu) = b^{''}(\theta) \text{为指数族分布的方差函数.} \tag{21}$$

$$\mu = E(Y) \tag{22}$$
$$\eta = X^T\beta \tag{23}$$

线性预测量$\eta = X^T\beta$和Y的期望之间，通过一个单调的连接函数$g(\cdot)$联系在一起，即$g(\mu)=\eta$:

- $g(\mu)=ln\mu \quad (\mu>0)$
- $g(\mu)=ln\frac{\mu}{1-\mu} \quad 0<\mu<1$
- $g(\mu)=\Phi^{-1}(\mu) \quad 0<\mu<1$
- ...

Logistic的LinkFunction为$g(\mu)=ln\frac{\mu}{1-\mu} \quad 0<\mu<1$，$g(\mu_i)=\sum_{j=1}^{p}X_{ij}\beta_j, \quad i=1,\dots,n$

(2)Logistic回归

设$X_1,\dots,X_p$的n组值$X_i=(X_{i1},\dots,X_{ip})^T, i=1,\dots,n$ 在每个X_i处，对二值随机变量ξ_i进行m_i次观测，其中$\xi_i=1$表示发生事件A，$\xi_i=0$表示未发生事件A.

设$\xi_i=1$有k_i次，令Y_i表示$\{\xi_i=1\}$出现的频率，则$Y_i=\frac{k_i}{m_i}, i=1,\dots,n, k_i=0,1,2,...,m_i$.

设$Y_1,\dots,Y_n$相互独立，则$Y_i\sim B(m_i,\mu_i)/m_i, i=1,\dots,n$,即$X_i=Y_im_i\sim B(m_i,\mu_i)$.

$P(Y=y)=P(X=ym)=C_m^{my}\mu^{my}(1-\mu)^{m-my}=exp\{\frac{yln\frac{\mu}{1-\mu}+ln(1-\mu)}{1/m}+ln(C_m^{my})\}$.

$\Rightarrow \theta=ln\frac{\mu}{1-\mu}, \phi=\frac{1}{m}, a(\phi)=\phi=\frac{1}{m}, b(\theta)=-ln(1-\mu)=ln(1+e^{\theta}), c(y,\phi)=lnC_m^{my}$.

设$Y_i\sim f(y;\theta_i,\phi_i)=exp\{\frac{y\theta_i-b(\theta_i)}{a(\phi_i)}+c(y,\phi_i)\}$，则其对数似然函数为

$$lnL(\beta_1,\dots,\beta_p)=ln[\sum_{i=1}^{n}f(Y_i;\theta_i,\phi_i)]=\sum_{i=1}^{n}[\frac{Y_i\theta_i-b(\theta_i)}{a(\phi_i)}+c(Y_i,\phi_i)] \tag{24}$$

$$\Rightarrow\frac{\partial lnL(\beta_1,\dots,\beta_p)}{\partial\beta_r}=\frac{\partial\sum_{i=1}^{n}\frac{Y_i\theta_i-b(\theta_i)}{a(\phi_i)}}{\partial\beta_r}=0, \quad r=1,\dots,p \tag{25}$$

$$\Rightarrow\sum_{i=1}^{n}\frac{(Y_i-\mu_i)X_{ir}}{a(\phi_i)V(\mu_i)g^{'}(\mu_i)}=0, \quad \mu_i=g^{-1}(\sum_{j=1}^{p}X_{ij}\beta_j), \quad r=1,\dots,p \tag{26}$$

(3)求解方法

$$\sum_{i=1}^{n}\frac{(Y_i-\mu_i)X_{ir}}{a(\phi_i)V(\mu_i)g^{'}(\mu_i)}=0, \quad \mu_i=g^{-1}(\sum_{j=1}^{p}X_{ij}\beta_j), \quad r=1,\dots,p$$

需通过迭代法来求解β

IRWLS

设$Y_i \sim N(\mu_i, \sigma_i^2), \sigma_i^2 = \sigma^2 a_i$且$a_1 \ldots a_n$已知，$Y_i$与$X_1, \ldots, X_p$服从线性模型，则

$$\mu_i = E(Y_i) = g(\mu_i) = \sum_{j=1}^{p} X_{ij}\beta_j \tag{27}$$

$$g(\mu) = \mu, V(\mu) = b^{''}(\theta) = 1, a(\phi) = a_i\phi = a_i\sigma^2 = \sigma_i^2, b(\theta) = \frac{\theta^2}{2}, \theta = \mu \tag{28}$$

$$\Rightarrow \sum_{i=1}^{n} \frac{(Y_i - \mu_i)X_{ir}}{a(\phi_i)V(\mu_i)g^{'}(\mu_i)} = \sum_{i=1}^{n} \frac{Y_i - \mu_i}{a_i} X_{ir} = 0 \tag{29}$$

$$Y = (y_1, y_2, \ldots, y_n)^T, X = \begin{bmatrix} X_{11} & X_{12} & \ldots & X_{1p} \\ X_{21} & X_{22} & \ldots & X_{2p} \\ \ldots & \ldots & \ldots & \ldots \\ X_{n1} & X_{n2} & \ldots & X_{np} \end{bmatrix}_{n\times p}, \tag{30}$$

$$\beta = (\beta_1, \ldots, \beta_p)^T, W = Diag(\frac{1}{a_1}, \ldots, \frac{1}{a_n}) \tag{31}$$

$$\Rightarrow \hat{\beta} = (X^TWX)^{-1}X^TWY \tag{32}$$

令$Z_i = g(\mu_i) + (Y_i - \mu_i)g^{'}(\mu_i)$，则$E(Z_i) = g(\mu_i), Var(Z_i) = a_i\phi V(\mu_i)[g^{'}(\mu_i)]^2 = \tilde{a}_i\phi.$

$\Rightarrow$线性模型$E(Z_i) = \sum_{j=1}^{p} X_{ij}\beta_j$的似然方程为$\sum_{i=1}^{n} \frac{Z_i - g(\mu_i)}{\tilde{a}_i} X_{ir} = \sum_{i=1}^{n} \frac{Y_i - \mu_i}{a_i V(\mu_i)} \frac{X_{ir}}{g^{'}(\mu_i)} = 0.$

$\Rightarrow \hat{\beta} = (X^TWX)^{-1}X^TWZ$

$$\eta_i^{(t)} = g(\mu_i^{(t)}) = \sum_{j=1}^{p} X_{ij}\beta_j^{(t)} \tag{33}$$

$$Z_i^{(t)} = \eta_i^{(t)} + (Y_i - \mu_i^{(t)})g^{'}(\mu_i^{(t)}) \tag{34}$$

$$W_i^{(t)} = \frac{1}{a_i V(\mu_i^{(t)})[g^{'}(\mu_i^{(t)})]^2} \tag{35}$$

$$\hat{\beta}^{(t+1)} = (X^TW^{(t)}X)^{-1}X^TW^{(t)}Z^{(t)} \tag{36}$$

IRWLS迭代步骤：

1. 给定一组初值$\mu_1^{(0)}, \ldots, \mu_n^{(0)}$，如$\mu_i^{(0)} = Y_i$
2. 计算$\eta_i^{(0)} = g(\mu_i^{(0)})$，Logistic中连接函数$g(\mu) = ln\frac{\mu}{1-\mu} \quad 0 < \mu < 1$
3. 计算$Z_i^{(0)} = \eta_i^{(0)} + (Y_i - \mu_i^{(0)})g^{'}(\mu_i^{(0)})$
4. 计算$W_i^{(0)} = \frac{1}{a_i V(\mu_i^{(0)})[g^{'}(\mu_i^{(0)})]^2}$

5. 求出$\hat{\beta}^{(1)} = (X^T W^{(0)} X)^{-1} X^T W^{(0)} Z^{(0)}$，其中$Z^{(0)} = (Z_1^{(0)}, Z_2^{(0)}, \ldots, Z_n^{(0)})^T$，$W^{(0)} = Diag(W_1^{(0)}, W_2^{(0)}, \ldots, W_n^{(0)})$
6. 令$\eta^{(1)} = (\eta_1^{(1)}, \eta_2^{(1)}, \ldots, \eta_n^{(1)})^T = X\hat{\beta}^{(1)}$，进而求出$Z_i^{(1)}, W_i^{(1)}, \hat{\beta}^{(1)}$
7. 迭代至$\hat{\beta}^{(t+1)}$收敛

在Logistic回归中，$g(\mu) = ln\frac{\mu}{1-\mu} \quad 0 < \mu < 1$，$g^{'}(\mu) = \frac{1}{\mu(1-\mu)}$，$V(\mu) = b^{''}(\theta) = \mu(1-\mu)$，代入IRWLS迭代步骤，即可求出Logistic模型中的估计系数$\hat{\beta}$。

Newton-Raphson迭代法与Fisher得分法

对数似然函数$f(\beta) = lnL(\beta_1, \ldots, \beta_p) = \sum_{i=1}^n [\frac{Y_i\theta_i - b(\theta_i)}{a(\phi_i)} + c(Y_i, \phi_i)]$

$\mu_i = b^{'}(\theta_i), g(\mu_i) = \sum_{j=1}^p X_{ij}\beta_j$

求解$\hat{\beta} = \arg\max_\beta f(\beta)$

Newton-Raphson迭代法：

- 设$f(\beta)$是$\beta = (\beta_1, \ldots, \beta_p)^T$的$p$元函数，求$\hat{\beta}$使$f(\hat{\beta}) = \max_\beta f(\beta)$
- 令$q = (q_1, \ldots, q_p)^T = (\frac{\partial f(\beta)}{\partial \beta_1}, \ldots, \frac{\partial f(\beta)}{\partial \beta_p})^T = \frac{\partial f(\beta)}{\partial \beta}, H = (h_{kl})_{p\times p} = \frac{\partial^2 f(\beta)}{\partial\beta\partial\beta^T}, h_{kl} = \frac{\partial^2 f(\beta)}{\partial\beta_l\beta_k}, 0 \le k, l \le p$
- $q^{(t)} = \frac{\partial f(\beta)}{\partial\beta}|_{\beta=\beta^{(t)}}, H^{(t)} = (h_{kl}^{(t)}), h_{kl}^{(t)} = \frac{\partial^2 f(\beta)}{\partial\beta_l\beta_k}|_{\beta=\hat{\beta}^{(t)}}$
- $f(\beta)$在$\beta = \hat{\beta}^{(t)}$处的二次Taylor展开：$f(\beta) \approx Q^{(t)}(\beta) = f(\beta^{(t)}) + (q^{(t)})^T(\beta - \beta^{(t)}) + \frac{1}{2}(\beta - \beta^{(t)})^T H^{(t)}(\beta - \beta^{(t)})$
- 令$\frac{\partial Q^{(t)}(\beta)}{\partial\beta} = q^{(t)} + H^{(t)}(\beta - \beta(t)) = 0, \ \Rightarrow \hat{\beta}^{(t+1)} = \beta^{(t)} - [H^{(t)}]^{-1}q^{(t)}, \quad t = 0, 1, 2, \ldots$

求多元函数极值问题的Newton-Raphson公式为

$$\hat{\beta}^{(t+1)} = \beta^{(t)} - [H^{(t)}]^{-1}q^{(t)}, \quad t = 0, 1, 2, \ldots \tag{N-R}$$

设指数族分布中的$a(\phi_i) = a_i\phi$且$a_i \ldots a_n$已知

- 得分向量Score Vector：$f(\beta)$关于$\beta_1, \ldots, \beta_p$的一阶偏导数所成的向量

$$\frac{\partial f(\beta)}{\partial \beta_k} = \frac{\partial lnL(\beta_1,\ldots,\beta_p)}{\partial \beta_k} = \sum_{i=1}^{n} \frac{Y_i-\mu_i}{a_i\phi V(\mu_i)g'(\mu_i)}X_{ik}, \quad k=1,\ldots,p \tag{37}$$

$$\mu_i = g^{-1}(\sum_{j=1}^{p} X_{ij}\beta_j), \quad i=1,\ldots,n \tag{38}$$

$$ScoreVector: \quad S(\beta;Y) = (S_1(\beta;Y),\ldots,S_p(\beta;Y))^T = (\frac{\partial f(\beta)}{\partial \beta_1},\ldots,\frac{\partial f(\beta)}{\partial \beta_p})^T \tag{39}$$

- 信息阵I：是Newton_Raphson迭代中的Hessian矩阵H的负矩阵

$$\text{信息阵各元素} I_{kl}(\beta;Y) = -\frac{\partial^2 f(\beta)}{\partial \beta_l \partial \beta_k} = -\frac{\partial}{\partial \beta_l}(\frac{\partial f(\beta)}{\partial \beta_k}) = -\frac{\partial}{\partial \beta_l}(\sum_{i=1}^{n} \frac{Y_i-\mu_i}{a_i\phi V(\mu_i)g'(\mu_i)}X_{ik}) \tag{40}$$

$$= -\sum_{i=1}^{n}[\frac{Y_i-\mu_i}{a_i\phi}\frac{\partial}{\partial \beta_l}(\frac{X_{ik}}{V(\mu_i)g'(\mu_i)}) - \frac{X_{ik}X_{il}}{a_i\phi V(\mu_i)[g'(\mu_i)]^2}] \tag{41}$$

$$I(\beta;Y) = (I_{kl}(\beta;Y))_{p\times p} = -H \tag{42}$$

- Fisher信息阵：是信息阵I的期望

$$\text{Fisher Information Matrix: } F(\beta) = E(I(\beta;Y)) = E(I_{kl}(\beta;Y)) = -E(H) \tag{43}$$

$$\Rightarrow E(I_{kl}(\beta;Y)) = \sum_{i=1}^{n} \frac{X_{ik}X_{il}}{a_i\phi V(\mu_i)(g'(\mu_i))^2} = \frac{1}{\phi}\sum_{i=1}^{n} W_iX_{ik}X_{il},\ W_i = \frac{1}{a_iV(\mu_i)(g'(\mu_i))^2} \tag{44}$$

$$\Rightarrow E(I_{kl}(\beta;Y)) = \frac{1}{\phi}(X^TWX)_{kl},\ W = Diag(W_1,\ldots,W_n),\ \text{与IRWLS中的}W\text{相同} \tag{45}$$

$$\Rightarrow F(\beta) = \frac{1}{\phi}(X^TWX)_{kl}, \quad F^{-1}(\beta) = \phi(X^TWX)^{-1} \tag{46}$$

- GLM中的Newton-Raphson迭代法

$$\hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} + I^{-1}(\hat{\beta}^{(t)};Y)S(\hat{\beta}^{(t)};Y),\ t=0,1,2,\ldots$$

I^{-1}中有因子ϕ，S中有因子$\frac{1}{\phi}$，故迭代公式与未知量ϕ无关.

- Fisher得分法(又称为“修正的N-R法”)：用“Fisher信息阵F”代替“信息阵I”

$$\hat{\beta}^{(t+1)} = \hat{\beta}^{(t)} + F^{-1}(\hat{\beta}^{(t)};Y)S(\hat{\beta}^{(t)};Y),\ t=0,1,2,\ldots$$

可证明Fisher得分法与IRWLS法等价

将Logistic模型中

$$\theta = g(\mu) = ln\frac{\mu}{1-\mu} = X^T\beta \tag{47}$$

$$\phi = \frac{1}{m}, a(\phi) = \phi = \frac{1}{m} \tag{48}$$

$$b(\theta) = -ln(1-\mu) = ln(1+e^{\theta}) \tag{49}$$

$$c(y,\phi) = lnC_m^{my} \tag{50}$$

$$V(\theta) = b^{''}(\theta) = \frac{e^{\theta}}{(1+e^{\theta})^2} = \mu(1-\mu) \tag{51}$$

代入上述IRWLS、Newton-Raphson、Fisher得分法的迭代公式中，即可求出$\hat{\beta}$.

2. 直接定义Logistic模型

$$P(Y=1|X=x_i) = \frac{exp(\beta_0 + x_i^T\beta)}{1+exp(\beta_0 + x_i^T\beta)} = p_i \tag{52}$$

$$P(Y=0|X=x_i) = \frac{1}{1+exp(\beta_0 + x_i^T\beta)} = 1-p_i \tag{53}$$

$$logit(p) = ln\frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p \tag{54}$$

$$\theta = (\beta_0, \beta^T)^T \tag{55}$$

$$L(\theta) = ln\prod_{i=1}^{n}[p(x_i)]^{y_i}[1-p(x_i)]^{1-y_i} \tag{56}$$

$$= \sum_{i=1}^{n}[y_i ln(p_i) + (1-y_i)ln(1-p_i)] \tag{57}$$

$$= \sum_{i=1}^{n}[y_i ln(\frac{p_i}{1-p_i}) + ln(1-p_i)] \tag{58}$$

$$= \sum_{i=1}^{n}[y_i(\beta_0 + x_i^T\beta) + ln(1-\frac{e^{\beta_0+x_i^T\beta}}{1+e^{\beta_0+x_i^T\beta}})] \tag{59}$$

$$= \sum_{i=1}^{n}[y_i(\beta_0 + x_i^T\beta) - ln(1+e^{\beta_0+x_i^T\beta})] \tag{60}$$

$$\Rightarrow \begin{cases} \frac{\partial ln[L(\theta)]}{\partial\beta_0} = \sum_{i=1}^{n}[y_i - \frac{e^{\beta_0+x_i^T\beta}}{1+e^{\beta_0+x_i^T\beta}}] \\ \frac{\partial ln[L(\theta)]}{\partial\beta} = \sum_{i=1}^{n}[y_i - \frac{e^{\beta_0+x_i^T\beta}}{1+e^{\beta_0+x_i^T\beta}}]x_i \end{cases} \tag{61}$$

参考资料

[1] https://liushulun.cn/post/machinelearning/ml-logistic/data-ml-logistic-optimization/ml-logistic-optimization/

[2] https://blog.csdn.net/lipengcn/article/details/52698895（LBFGS）

[3] https://liuxiaofei.com.cn/blog/lbfgs方法推导/#lbfgs方法推导（LBFGS）

[4] https://www.cnblogs.com/pinard/p/6018889.html（坐标轴下降法与最小角回归法）

[5] https://www.cs.cmu.edu/~ggordon/10725-F12/slides/（PPT优化方法）