1.6 线性回归模型
——优化算法

CSDN学院

2017年10月

►线性回归

- 模型
  - 目标函数（损失函数、正则）
    - 概率解释
- 优化求解
- 模型选择

▶ 线性回归的目标函数

- 无正则的最小二乘线性回归（Ordinary Least Square，OLS）

$$J(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - \mathbf{w}^T \mathbf{x}_i \right)^2$$

- L2正则的岭回归（Ridge Regression）模型：

$$J(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - \mathbf{w}^T \mathbf{x}_i \right)^2 + \lambda \|\mathbf{w}\|_2^2$$

- L1正则的Lasso模型：

$$J(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - \mathbf{w}^T \mathbf{x}_i \right)^2 + \lambda |\mathbf{w}|$$

▶ 模型训练

- 模型训练：根据训练数据求目标函数取极小值的参数

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} J(\mathbf{w})$$

- 目标函数极小值
  - 一阶的导数为0：$\dfrac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0$
  - 二阶导数>0：$\dfrac{\partial J^2(\mathbf{w})}{\partial \mathbf{w}^2} > 0$

▶ OLS的优化求解

CSDN 不止于代码

- 将OLS的目标函数写成矩阵形式

$$J(\mathbf{w}) = \sum_{i=1}^{N}\left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2 = (\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})$$

- 只取与w有关的项，得到

$$J(\mathbf{w}) = \mathbf{w}^T\left(\mathbf{X}^T\mathbf{X}\right)\mathbf{w} - 2\mathbf{w}^T\left(\mathbf{X}^T\mathbf{y}\right)$$

$$\boxed{\begin{aligned}&\frac{\partial}{\partial \mathbf{y}}\left(\mathbf{y}^T\mathbf{A}\mathbf{y}\right) = \left(\mathbf{A}+\mathbf{A}^T\right)\mathbf{y}\\ &\frac{\partial}{\partial \mathbf{a}}\left(\mathbf{b}^T\mathbf{a}\right) = \mathbf{b}\end{aligned}}$$

- 求导

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y} = 0 \quad \Rightarrow \quad \mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$\boxed{\hat{\mathbf{w}}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

(ordinary least squares, OLS)

通常通过奇异值分解（singular value decomposition，SVD）求解

▶ OLS的计算

- SVD 分解

$$\min\left(\left(ND^2+D^3\right),\left(DN^2+N^3\right)\right)$$

$$\mathbf{y}=\mathbf{X}\mathbf{w}+\varepsilon$$

令$\boldsymbol{\beta}=\mathbf{V}^T\mathbf{w}$

$$=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{w}+\varepsilon=\mathbf{U}\boldsymbol{\Sigma}\boldsymbol{\beta}+\varepsilon$$

$\boxed{\mathbf{X}=\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T}$ $\boxed{\mathbf{X}^T=\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T}$

$\boxed{\mathbf{X}^T\mathbf{X}=\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T=\boldsymbol{\Sigma}^2}$

$$\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}}=\boldsymbol{\Sigma}\mathbf{V}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \leftarrow \boxed{\hat{\mathbf{w}}_{mle}:=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

$$=\boldsymbol{\Sigma}\mathbf{V}^T\left(\boldsymbol{\Sigma}^2\right)^{-1}\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{y}$$

$$=\mathbf{U}^T\mathbf{y}$$

$\boxed{\mathbf{U}^T\mathbf{U}=\mathbf{I}_N}$ 列正交

$$\boldsymbol{\Sigma}\hat{\boldsymbol{\beta}}=\mathbf{U}^T\mathbf{y} \qquad \Rightarrow \hat{\boldsymbol{\beta}}=\boldsymbol{\Sigma}^{-1}\mathbf{U}^T\mathbf{y}$$

$$\mathbf{V}^T\hat{\mathbf{w}}=\hat{\boldsymbol{\beta}}=\boldsymbol{\Sigma}^{-1}\mathbf{U}^T\mathbf{y}$$

$$\boxed{\hat{\mathbf{w}}=\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^T\mathbf{y}}$$

$\boxed{\mathbf{V}\mathbf{V}^T=\mathbf{V}^T\mathbf{V}=\mathbf{I}_D}$ 行、列均正交

6

▶ 岭回归的优化求解

CSDN 不止于代码

- 岭回归的目标函数与OLS只相差一个正则项（也是w的二次函数），所以类似可得：

$$J(\mathbf{w})=\sum_{i=1}^{N}\left(y_i-\mathbf{w}^T\mathbf{x}_i\right)^2+\lambda\|\mathbf{w}\|_2^2=(\mathbf{y}-\mathbf{X}\mathbf{w})^T(\mathbf{y}-\mathbf{X}\mathbf{w})+\lambda\mathbf{w}^T\mathbf{w}$$

- 求导，得到

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}=2\mathbf{w}^T\left(\mathbf{X}^T\mathbf{X}\right)\mathbf{w}-2\mathbf{w}^T\left(\mathbf{X}^T\mathbf{y}\right)+2\lambda\mathbf{w}^T=0$$

$$\boxed{\hat{\mathbf{w}}_{ridge}=\left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}_D\right)^{-1}\mathbf{X}^T\mathbf{y}}$$

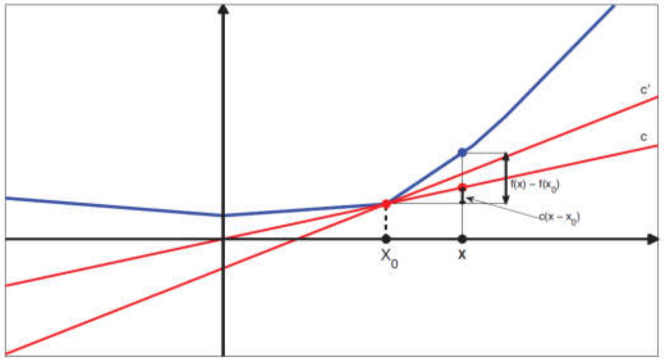
Lasso的优化条件

- Lasso的目标函数为$J(\mathbf{w},\lambda)=RSS(\mathbf{w})+\lambda\|\mathbf{w}\|_1$
- 但项 $\|\mathbf{w}\|_1$ 在 $w_j=0$ 处不可微（不平滑优化问题）
- 为了处理不平滑函数，扩展导数的表示，定义一个(凸)函数f在点x_0处的次梯度(subgradient)或次导数(subderivative)为一个标量g，使得

$$f(x)-f(x_0)\ge g(x-x_0),\forall x\in\mathcal{I}$$

- 其中$\mathcal{I}$为包含x_0的某个区间。
- 定义区间[a,b]的子梯度集合为

$$a=\lim_{x\to x_0^-}\frac{f(x)-f(x_0)}{x-x_0},b=\lim_{x\to x_0^+}\frac{f(x)-f(x_0)}{x-x_0}$$



▶ Lasso的优化条件

- 所有次梯度的区间称为函数 f 在 x_0 处的次微分(subdifferential)，用 $\partial f(x)|_{x_0}$ 表示
- 例：绝对值函数 $f(x)=|x|$，其次梯度为

$$\partial f(x)=\begin{cases}\{-1\} & if\ \ x<0\\ [-1,+1] & if\ \ x=0\\ \{+1\} & if\ \ x>0\end{cases}$$

- 如果函数处处可微，$\partial f(x)=\left\{\dfrac{df(x)}{dx}\right\}$
- 同标准的微积分类似，可以证明当且仅当 $0\in\partial f(x)|_{\hat{x}}$ 时，为 f 的局部极值点。

▶ Lasso的优化条件

- 将上述结论带入Lasso问题
- 目标函数为：$J(\mathbf{w},\lambda)=RSS(\mathbf{w})+\lambda\|\mathbf{w}\|_1=\sum_{i=1}^{N}\left(y_i-\mathbf{w}_i^T\mathbf{x}_i\right)^2+\lambda\|\mathbf{w}\|_1$
- 可微项的梯度：

$$\frac{\partial}{\partial w_j}RSS(\mathbf{w})=\frac{\partial}{\partial w_j}\sum_{i=1}^{N}\left(y_i-\left(\mathbf{w}_{-j}^T\mathbf{x}_{i,-j}+w_jx_{ij}\right)\right)^2$$

$$=-2\sum_{i=1}^{N}\left(y_i-\mathbf{w}_{-j}^T\mathbf{x}_{i,-j}-w_jx_{ij}\right)x_{ij}$$

$$=2\sum_{i=1}^{N}x_{ij}^2w_j-2\sum_{i=1}^{N}\left(y_i-\mathbf{w}_{-j}^T\mathbf{x}_{i,-j}\right)x_{ij}$$

$$=a_jw_j-c_j$$

$$a_j=2\sum_{i=1}^{N}x_{ij}^2$$

$$\boxed{c_j=2\sum_{i=1}^{N}x_{ij}\left(\boxed{y_i-\mathbf{w}_{-j}^T\mathbf{x}_{i,-j}}\right)}$$

第j维特征与残差的相关性

利用D-j 维特征得到的预测的残差

▶ Lasso的优化条件

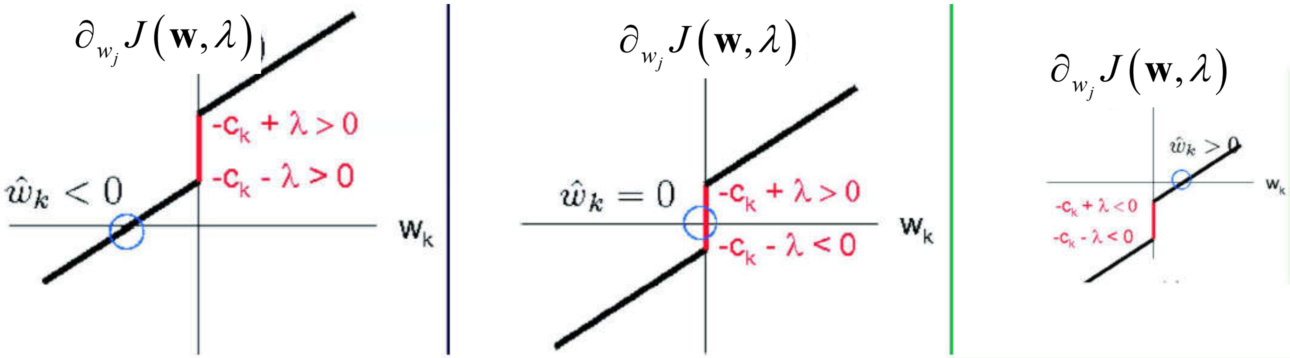
- 目标函数的子梯度(subgradient)

$$\partial_{w_j} J(\mathbf{w},\lambda) = \left(a_j w_j - c_j\right) + \lambda \partial_{w_j} \|\mathbf{w}\|_1$$

$$= \begin{cases} \{a_j w_j - c_j - \lambda\} & if \;\; w_j < 0 \\ \left[-c_j - \lambda, -c_j + \lambda\right] & if \;\; w_j = 0 \\ \{a_j w_j - c_j + \lambda\} & if \;\; w_j > 0 \end{cases}$$

▶ Lasso的优化条件

- 根据c_j的不同，$\partial_{w_j} J(\mathbf{w},\lambda)=0$有三种情况



$c_j < -\lambda, \hat{w}_j = \dfrac{c_j+\lambda}{a_j}$
特征与残差强负相关

$\hat{w}_j = 0$
特征与残差弱相关('+/-')

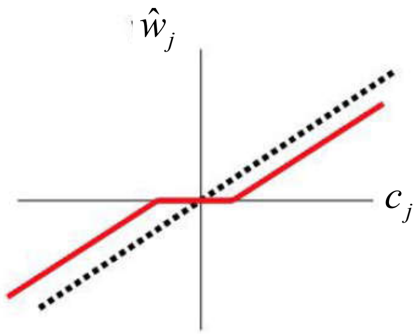
$\hat{w}_j = \dfrac{c_j-\lambda}{a_j}$
特征与残差强正相关

▶ 软 & 硬阈值

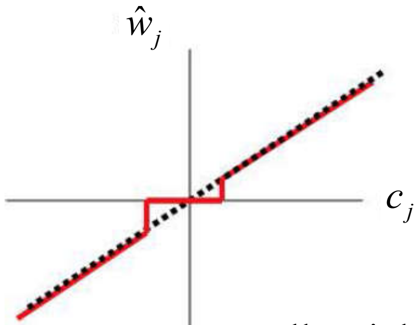
- 软 & 硬阈值

$$\hat{w}_j(c_j)=\begin{cases}(c_j+\lambda)/a_j & if\ c_j<-\lambda\\ 0 & if\ c_j\in[-\lambda,\lambda]\\ (c_j-\lambda)/a_j & if\ c_j>\lambda\end{cases}=\mathrm{soft}\left(\frac{c_j}{a_j};\frac{\lambda}{a_j}\right)$$

$$\mathrm{soft}(a;\delta)=\mathrm{sign}(a)(|a|-\delta)_+$$



Soft: set small weights to zero and "shrink" all other weights.
Biased estimator



Hard: set small weights to zero without "shrinking" all other weights.
Unbiased estimator

- 预计算 $a_j = 2\sum_{i=1}^{N} x_j^2$
- 初始化参数**w**（全0或随机）
  循环直到收敛:
  - for $j = 0,1,\ldots D$
    - 计算 $c_j = 2\sum_{i=1}^{N} x_j (y_i - \mathbf{w}_{-j}^T \mathbf{x}_{i,-j})$
    - 更新w_j : $\hat{w}_j\left(c_j\right) = \begin{cases} \left(c_j + \lambda\right)/a_j & if\ c_j < -\lambda \\ 0 & if\ c_j \in [-\lambda, \lambda] \\ \left(c_j - \lambda\right)/a_j & f\ c_j > \lambda \end{cases} = \mathrm{soft}\left(\frac{c_j}{a_j}; \frac{\lambda}{a_j}\right)$
  - 选择变化幅度最大的维度进行更新

坐标轴下降法

- 为了找到一个函数的局部极小值，在每次迭代中可以在当前点处沿一个坐标方向进行一维搜索。
- 整个过程中循环使用不同的坐标方向。一个周期的一维搜索迭代过程相当于一个梯度迭代。

- 注意：
  - 梯度下降方法是利用目标函数的导数（梯度）来确定搜索方向的，而该梯度方向可能不与任何坐标轴平行。
  - 而坐标轴下降法是利用当前坐标系统进行搜索，不需要求目标函数的导数，只按照某一坐标方向进行搜索最小值。（在稀疏矩阵上的计算速度非常快，同时也是Lasso回归最快的解法）

▶ 梯度下降

- 在线性回归中，目标函数中的训练误差部分为

$$J(\mathbf{w})=\sum_{i=1}^{N}\left(f(\mathbf{x}_i)-y_i\right)^2$$

- 梯度为

$$g(\mathbf{w})=\frac{\partial}{\partial \mathbf{w}}J(\mathbf{w})=\sum_{i=1}^{N}2\left(f(\mathbf{x}_i)-y_i\right)\mathbf{x}_i$$

- L2正则的梯度计算简单（L1正则需要计算次梯度）

$$\Omega(\mathbf{w})=\lambda\|\mathbf{w}\|_2^2=\lambda\mathbf{w}^T\mathbf{w},\quad \frac{\partial\Omega(\mathbf{w})}{\partial \mathbf{w}}=2\lambda\mathbf{w}$$

随机梯度下降 (Stochastic Gradient Descent, SGD)

- 在上述梯度下降算法中，梯度$g(\mathbf{w}) = \boxed{\sum_{i=1}^{N}} 2\left(f(\mathbf{x}_i) - y_i\right)\mathbf{x}_i$
  - 利用所有的样本，因此被称为“批处理梯度下降”
- 随机梯度下降：每次只用一个样本$(\mathbf{x}_t, y_t)$

$$g(\mathbf{w}) = 2\left(f(\mathbf{x}_t) - y_t\right)\mathbf{x}_t$$

  - 通常收敛会更快，且不太容易陷入局部极值
  - 亦被称为在线学习(Online Learning)
  - 对大样本数据集尤其有效
- SGD的变形：
  - 可用于离线学习：每次看一个样本，对所有样本可重复多次循环使用（一次循环称为一个epoch）
  - 每次可以不止看一个样本，而是看一些样本（mini-batch)

小结

- 线性回归模型比较简单
  - 当数据规模较小时，可直接解析求解
    - scikit learn中的实现采用SVD分解实现
  - 当数据规模较大时，可采用随机梯度下降
    - Scikit learn提供一个SGDRegression类
- 岭回归求解类似OLS，采用SVD分解实现
- Lasso优化求解采用坐标轴下降法