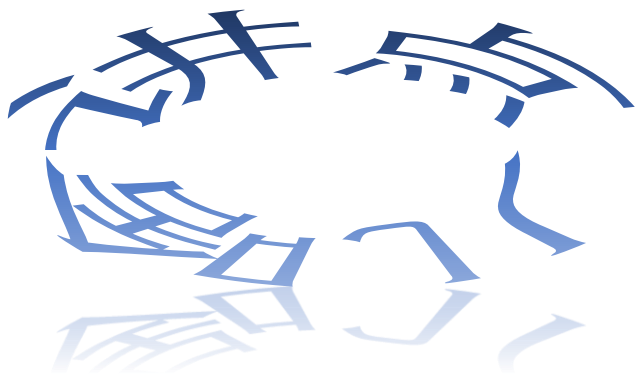
第3章 线性回归

电子工程学院、人工智能学院
college of Electronic Engineering、college of Artificial Intelligence



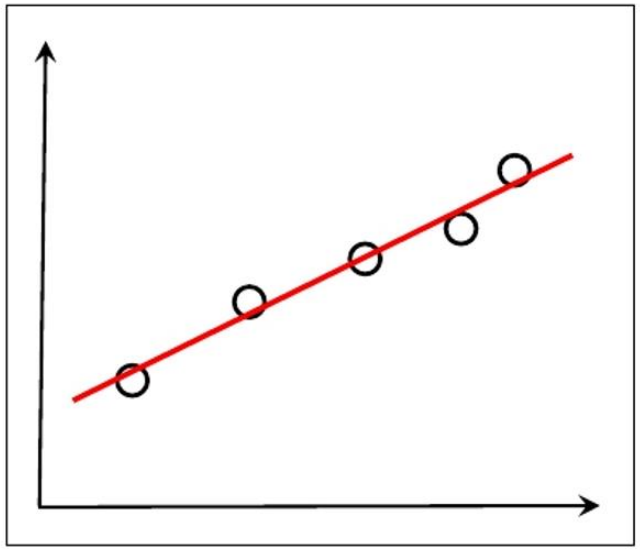
3.1 线性回归的基本形式

3.2 回归任务的损失函数

3.3 线性回归模型的正则项

3.4 线性回归的的优化算法

3.5 回归任务的模型性能评价



3.1 线性回归的基本形式

➢ 回忆：机器学习的三要素

1. 函数集合$\{f_1, f_2, \dots\}$
2. 目标函数$J(f)$：函数的好坏
3. 优化算法：找到最佳函数

线性回归：函数集合为输入特征的线性组合，即假设输出y与输入$\boldsymbol{x}$之间的关系为线性关系

$$\hat{y} = f(\boldsymbol{x}) = \underbrace{w_1 x_1}_{\text{权重}} + \underbrace{w_2 x_2}_{\text{权重}} + \dots + \underbrace{w_d x_d}_{\text{权重}} + \underbrace{b}_{\text{截距项}}$$

- d: 特征维数
- $\boldsymbol{x} = (1, x_1, \dots, x_d)^{\mathrm{T}}$：在$d$维特征的基础上，增加一个常数项1（用于表示截距项）

向量形式：

$$\hat{y} = f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} + b$$

$$\boldsymbol{w} = (w_1; w_2; \dots; w_d)$$

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \qquad D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$$

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ x_{21} & x_{22} & \cdots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} & 1 \\ \boldsymbol{x}_2^{\mathrm{T}} & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_m^{\mathrm{T}} & 1 \end{pmatrix} \qquad \boldsymbol{y} = (y_1; y_2; \ldots; y_m)$$

$$\boxed{f(\boldsymbol{x}_i) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b \text{ 使得 } f(\boldsymbol{x}_i) \simeq y_i}$$

$$\boldsymbol{x}_i = (x_{i1}; x_{i2}; \ldots; x_{id}) \qquad y_i \in \mathbb{R}$$

一个例子

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

离散属性处理

- 有“序”关系

  连续化为连续值

- 无“序”关系

  有k个属性值，则转化为k维向量

3.2 回归任务的损失函数

- 回忆：机器学习的三要素
  1. 函数集合$\{f_1, f_2, \dots\}$
  2. 目标函数$J(f)$：函数的好坏
  3. 优化算法：找到最佳函数

回归任务目标函数

$$J(f,\lambda) = \sum_{i=1}^{m} L(y_i, \hat{y}_i) + \lambda R(f)$$

损失函数　　正则项

3.2.1 均方差损失函数（ L2损失 ）

$$L(y,\hat{y}) = (y-\hat{y})^2 = r^2 = E$$

预测残差(Residual) $r = y - \hat{y}$

$$E_{\hat{w}} = (\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}})^{\mathrm{T}} (\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}})$$

3.2.2 L2损失函数的概率解释*

- 最小L2损失函数等价于高斯白噪声下的极大似然。
- 在回归任务中，令模型预测值和真实值之间的差异为噪声ε。假设噪声ε_i相互独立，且ε_i的分布为0均值的正态分布，即$\varepsilon_i \sim N(0, \sigma^2)$，则
- $y_i = f(\boldsymbol{x}_i) + \varepsilon_i$
- $y_i|\boldsymbol{x}_i \sim N(f(\boldsymbol{x}_i), \sigma^2)$
- 所以 $$p(y_i|\boldsymbol{x}_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y_i - f(\boldsymbol{x}_i)\right)^2}{2\sigma^2}\right)$$

◆ 补充：极大似然估计

- 给定数据$\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^m$，似然函数定义为数据出现的概率。
- 通常我们假定数据是独立同分布样本，因此所有数据出现的概率等于每个数据集中每个样本出现的概率相乘。
- log似然函数：

$$\ell(\boldsymbol{\theta}) = \ln p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{i=1}^{m} \ln p(\boldsymbol{x}_i|\boldsymbol{\theta})$$

其中$\boldsymbol{\theta}$为分布的参数。

- 极大似然估计，即$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta})$。

3.2.2 L2损失函数的概率解释*

- 单个数据点的概率密度函数为:

$$p(y_i|\boldsymbol{x}_i) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y_i - f(\boldsymbol{x}_i)\right)^2}{2\sigma^2}\right)$$

- 则log似然函数为

$$\ell(f) = \ln p(\mathcal{D}) = \sum_{i=1}^{m} \ln p(y_i|\boldsymbol{x}_i)$$

$$= \sum_{i=1}^{\mathrm{m}} \ln\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(y_i - f(\boldsymbol{x}_i)\right)^2}{2\sigma^2}\right)\right]$$

$$= -\frac{m}{2}\ln(2\pi) - \mathrm{m}\ln\sigma - \sum_{i=1}^{m}\frac{\left(y_i - f(\boldsymbol{x}_i)\right)^2}{2\sigma^2}$$

3.2.2 L2损失函数的概率解释*

log似然

$$\ell(f) = -\frac{m}{2}\ln(2\pi) - m\ln\sigma - \sum_{i=1}^{m}\frac{\left(y_i - f(\boldsymbol{x}_i)\right)^2}{2\sigma^2}$$

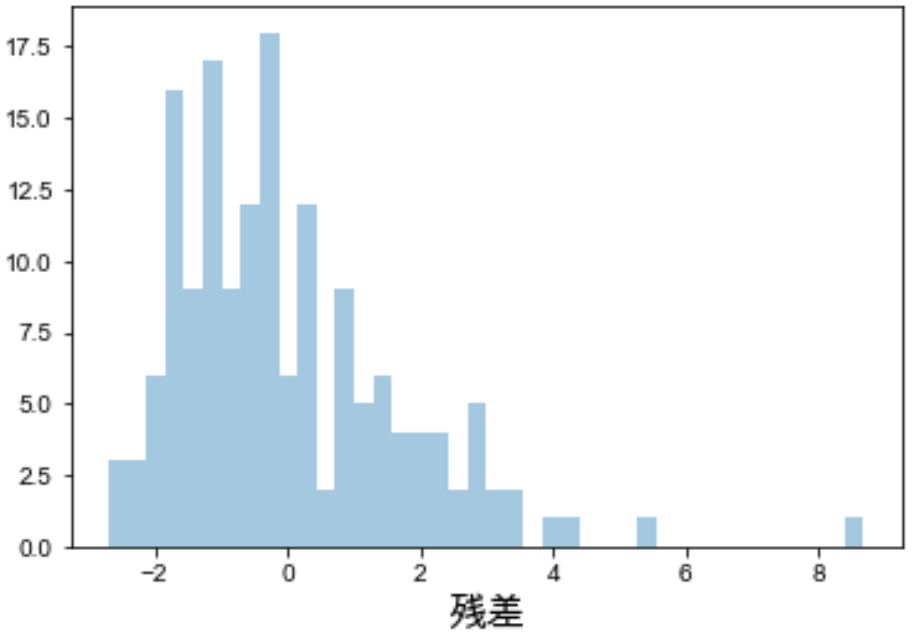
- 取最大值时，$-\sum_{i=1}^{m}\frac{\left(y_i-f(\boldsymbol{x}_i)\right)^2}{2\sigma^2}$取最大值，即$\sum_{i=1}^{m}\frac{\left(y_i-f(\boldsymbol{x}_i)\right)^2}{2\sigma^2}$取最小值。
- $\sum_{i=1}^{m}\left(y_i - f(\boldsymbol{x}_i)\right)^2$

所以：极大似然估计等价于最小L2损失函数

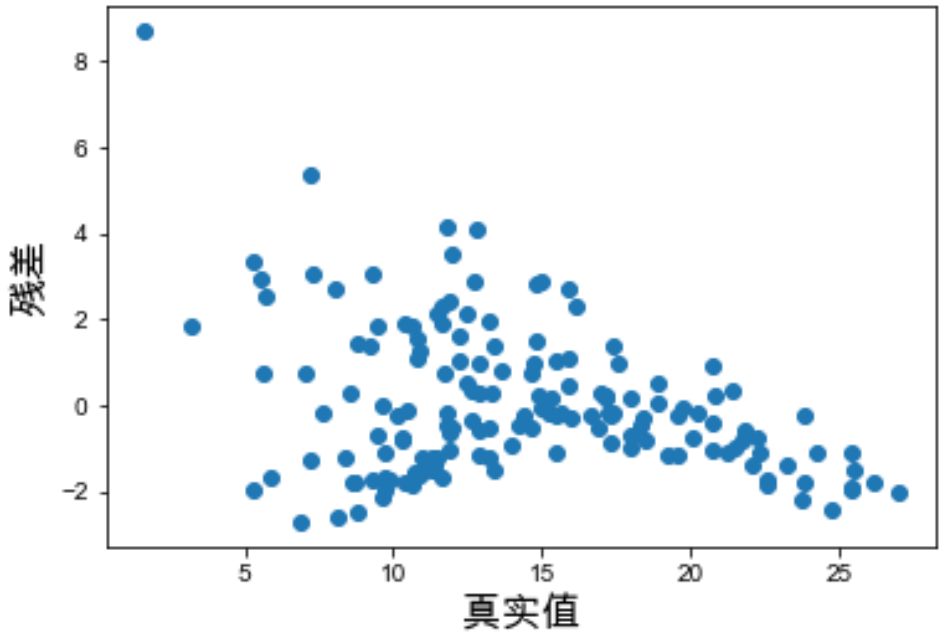
最小L2损失是高斯白噪声假设下的极大似然（负log似然损失）

最小L2损失是高斯白噪声假设下的极大似然（负log似然损失）

所以，可通过残差分布检验回归模型



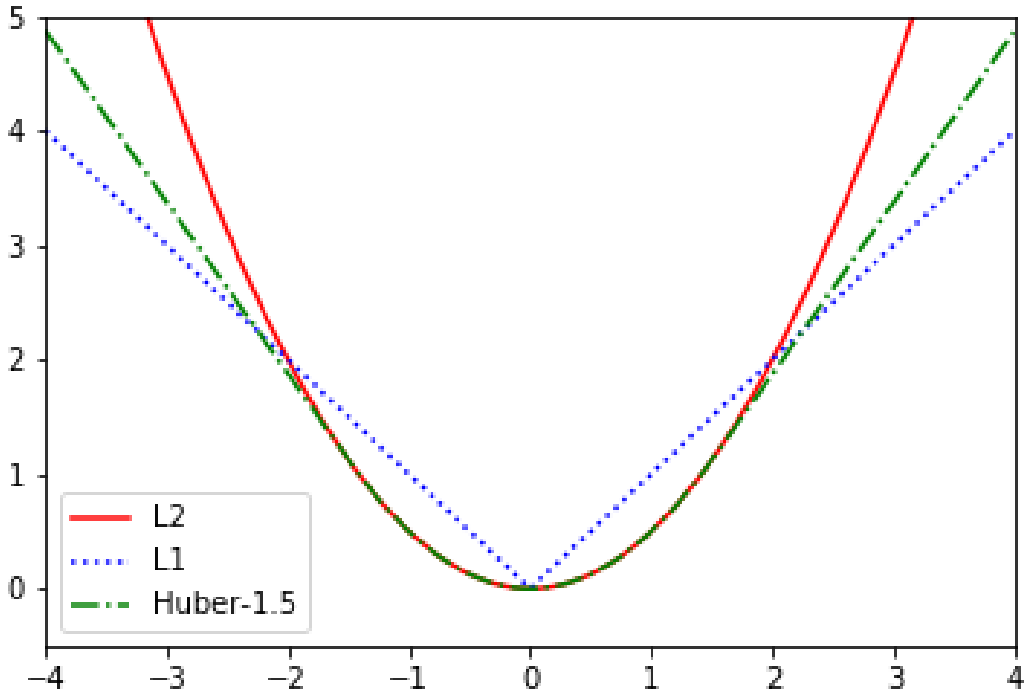
残差的分布并不符合0均值的正态分布
该模型（最小二乘回归模型）预测效果并不好



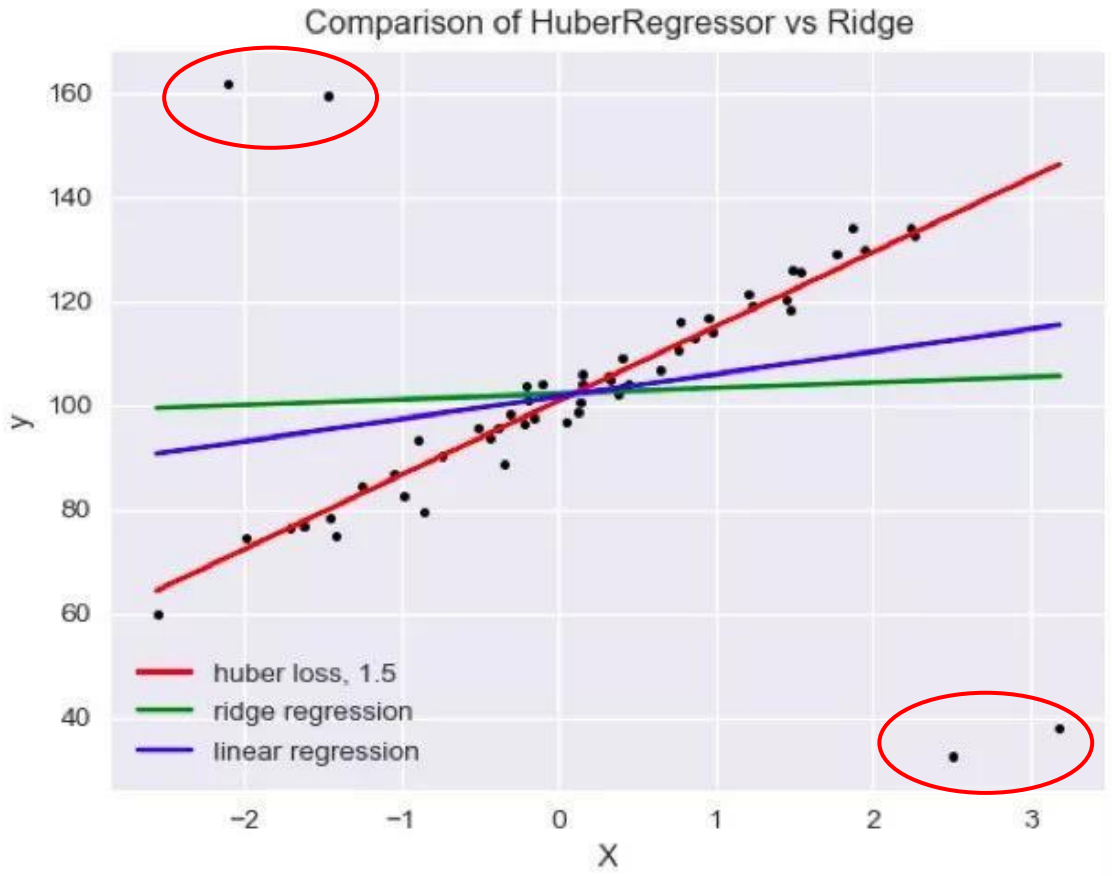
从真实值和残差的散点图来看，真实值较小和较大时，预测残差大多<0,其余情况残差大多>0。
模型还没有完全建模y与x之间的关系，还有一部分关系残留在残差中

3.2.3 胡伯损失函数（HuberRegressor）

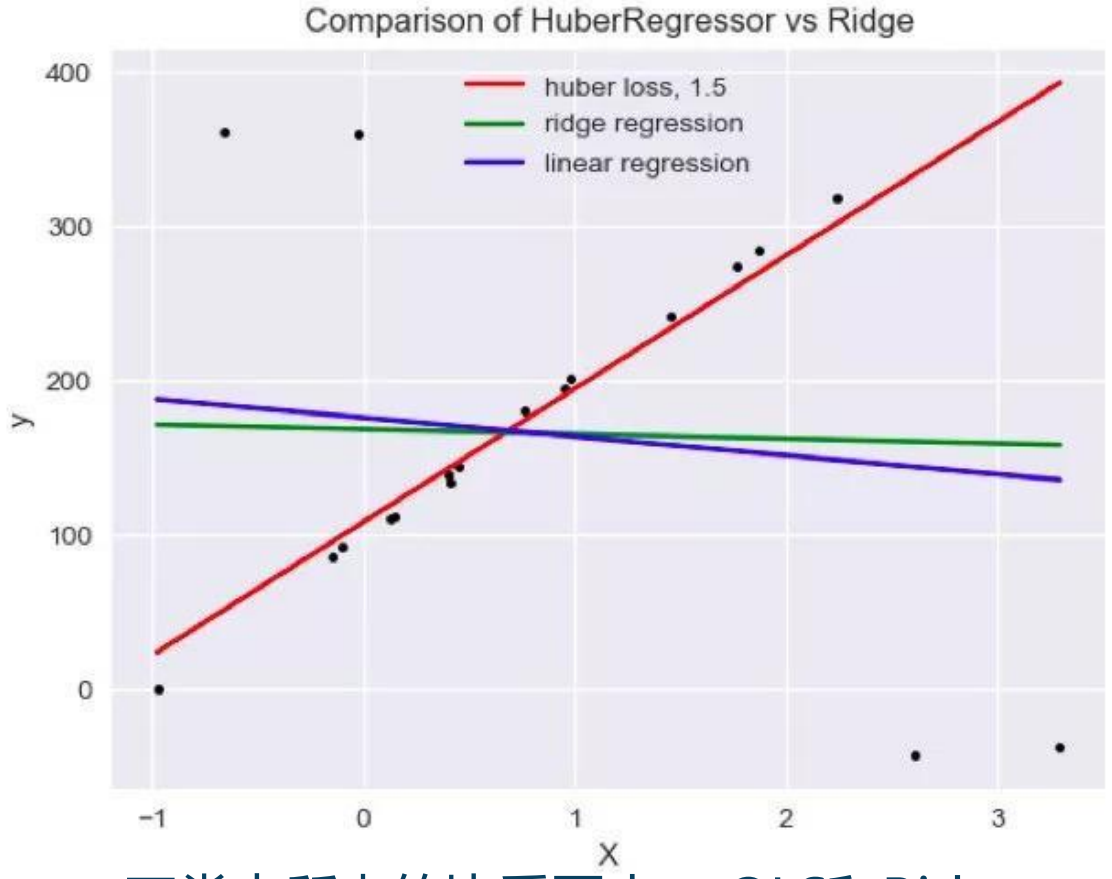
$$L_\delta(y,\hat{y}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & |r| \le \delta \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2 & otherwise \end{cases}$$



L2损失、L1损失和Huber损失比较
横轴：预测残差$r = y - \hat{y}$
纵轴：损失函数的值



左上方和右下出现了一些异常点
OLS和Ridge regression （L2损失）都不同程度上受到了异常点的影响，而Huber损失却没有受任何影响



正常点所占的比重更小， OLS和Ridge regression （L2损失）所决定出的回归模型几乎不工作，Huber损失性能依然完美

3.3 线性回归模型的正则项

- 回忆：机器学习的三要素
  1. 函数集合$\{f_1, f_2, \dots\}$
  2. 目标函数$J(f)$：函数的好坏
  3. 优化算法：找到最佳函数

回归任务的目标函数

$$J(f,\lambda) = \sum_{i=1}^{N} L(y_i, \hat{y}_i) + \lambda R(f)$$

损失函数　　正则项

抑制过拟合：在目标函数中加入正则项

正则项

$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$，正则项$R(f) = R(\boldsymbol{w})$

- 无正则
- L2正则：$R(\boldsymbol{w}) = \|\boldsymbol{w}\|_2^2 = \sum_{j=1}^{d} w_j^2$
- L1正则：$R(\boldsymbol{w}) = \|\boldsymbol{w}\|_1 = \sum_{j=1}^{d} |w_j|$

其中$\boldsymbol{w}$为模型参数，d为参数的维数。

注意：正则项不对截距项惩罚，因为截距项不影响模型的复杂度（函数的平滑程度：$\Delta\boldsymbol{x} \to \Delta y$）

3.3.1 无正则：最小二乘线性回归

$$J(\boldsymbol{w}) = \sum_{i=1}^{m} L(y_i, f(\boldsymbol{x}_i, \boldsymbol{w})) = \sum_{i=1}^{m} (y_i - f(\boldsymbol{x}_i, \boldsymbol{w}))^2 = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2$$

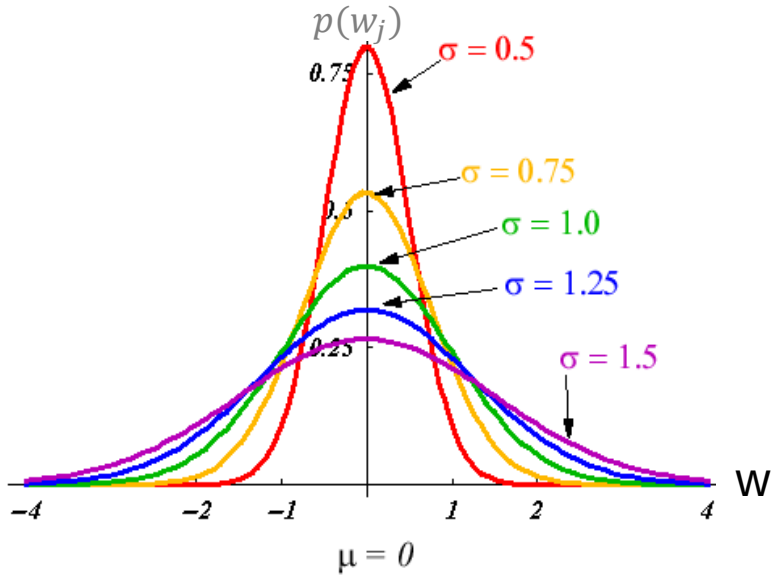
$$\hat{\boldsymbol{w}}^* = \arg\min_{\hat{\boldsymbol{w}}} (\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}})^{\mathrm{T}} (\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}) \qquad (3.9)$$

3.3.2 岭回归(Ridge Regression)：L2正则的线性回归

$$J(\boldsymbol{w},\lambda)=\sum_{i=1}^{m}L\big(y_i,f(\boldsymbol{x}_i,\boldsymbol{w})\big)+\lambda R(\boldsymbol{w})\ =\sum_{i=1}^{m}(y_i-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)^2+\lambda\sum_{j=1}^{D}w_j^2$$

$$=\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2+\lambda\|\boldsymbol{w}\|_2^2\ =(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\lambda\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$$

L2正则可视为参数先验分布为零均值正态分布



3.3.3 Lasso：L1正则的线性回归

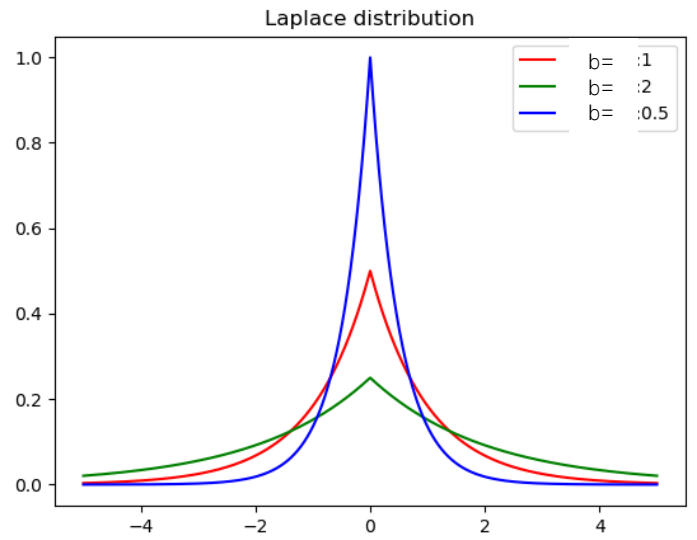
Lasso (Least Absolute Shrinkage and Selection Operator)的目标函数为：

$$J(\boldsymbol{w},\lambda)=\sum_{i=1}^{m}L\big(y_i,f(\boldsymbol{x}_i,\boldsymbol{w})\big)+\lambda R(\boldsymbol{w})$$

$$=\sum_{i=1}^{m}(y_i-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)^2+\lambda\sum_{j=1}^{D}|w_j|$$

$$=\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|^2+\lambda\|\boldsymbol{w}\|_1$$

L1正则可视为参数先验分布为Laplace分布



3.3.4 弹性网络：L1正则 + L2正则

- 正则项还可以为L1正则和L2正则的线性组合：

$$R(\boldsymbol{w},\rho)=\sum_{j=1}^{D}\left(\rho|w_j|+\frac{(1-\rho)}{2}w_j^2\right)$$

- 得到弹性网络的目标函数：

$$J(\boldsymbol{w},\lambda,\rho)=\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|^2+(\lambda_1\|\boldsymbol{w}\|_1+\lambda_2\|\boldsymbol{w}\|_2^2),\quad 0\le\rho\le 1.$$

3.4 线性回归模型的优化算法

- 回忆：机器学习的三要素
  1. 函数集合$\{f_1, f_2, \dots\}$
  2. 目标函数$J(f)$：函数的好坏
  3. 优化算法：找到最佳函数

- 解析求解（正规方程组）
- 梯度下降
- 坐标轴下降

3.4.1 解析求解——最小二乘回归（least square regression）

- 给定超参数λ的情况下，目标函数最优解：

$$\hat{\boldsymbol{w}} * = \underset{\boldsymbol{w}}{\operatorname{argmin}} J(\boldsymbol{w}, \lambda) = \underset{\boldsymbol{w}}{\operatorname{argmin}} (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}})^{\mathrm{T}} (\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{w}}) \qquad (3.9)$$

- 根据优化理论，函数极值点只能在边界点、不可导点、临界点（导数为0的点）
- 临界点：一阶偏导数组成的向量（亦被称为梯度）为0向量：

$$\frac{\partial J}{\partial \hat{\boldsymbol{w}}} = -2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\hat{\boldsymbol{w}} = \boldsymbol{0} \qquad (3.10)$$

$$\left[\frac{\partial(\boldsymbol{b}\boldsymbol{y})}{\partial \boldsymbol{y}} = \boldsymbol{b}^{\boldsymbol{T}}\right]$$

$$\left[\frac{\partial(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{b})}{\partial \boldsymbol{y}} = \boldsymbol{b}\right]$$

$$\left[\frac{\partial(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{y})}{\partial \boldsymbol{y}} = (A^{\mathrm{T}} + \boldsymbol{A})\boldsymbol{y}\right]$$

$$\frac{\partial J}{\partial \hat{\boldsymbol{w}}} = -2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + 2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\hat{\boldsymbol{w}} = \boldsymbol{0} \tag{3.10}$$

满秩讨论：

- $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$ 是满秩矩阵或正定矩阵，则

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \rightarrow \hat{\boldsymbol{w}}^{*}{}_{OLS} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} \tag{3.11}$$

- $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$ 不是满秩矩阵
  - 根据归纳偏好选择解
  - 引入正则化

3.4.3 梯度下降——最小二乘回归

最小二乘的目标函数：$J(\boldsymbol{w},\lambda)=(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})$

目标函数的梯度：$$g(\boldsymbol{w})=\frac{\partial J(\boldsymbol{w},\lambda)}{\partial \boldsymbol{w}}=-2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}+2(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})\boldsymbol{w}$$

$$=-2\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})$$ 预测残差r

参数更新：$$\boldsymbol{w}^{(t+1)}=\boldsymbol{w}^{(t)}-\eta g\left(\boldsymbol{w}^{(t)}\right)$$

$$=\boldsymbol{w}^{(t)}+2\eta\boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}^{(t)}\right)$$

参数的更新量与输入与预测残差的相关性有关。

3.4.4 梯度下降——岭回归

- 岭回归的目标函数：$J(\boldsymbol{w},\lambda)=(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})+\lambda\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$

- 目标函数的梯度：

$$g(\boldsymbol{w})=\frac{\partial J(\boldsymbol{w},\lambda)}{\partial \boldsymbol{w}}=-2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}+2(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})\boldsymbol{w}+2\lambda\boldsymbol{w}$$

- 参数更新：$\boldsymbol{w}^{(t+1)}=\boldsymbol{w}^{(t)}-\eta g\left(\boldsymbol{w}^{(t)}\right)$

$$=\boldsymbol{w}^{(t)}+2\eta\boldsymbol{X}^{\mathrm{T}}\left(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}^{(t)}\right)-2\eta\lambda\boldsymbol{w}^{(t)}$$

3.4.5 梯度下降进阶

✓ 梯度的计算：计算梯度时需用到每个训练样本，当样本数目很多时，计算费用高：随机梯度下降、小批量梯度下降

✓学习率：

- 太小，收敛慢
- 学习率太大，不收敛
- 各参数公用一个学习率：特征缩放

✓ 梯度计算:

机器学习算法的目标函数: $J(\boldsymbol{\theta}) = \sum_{i=1}^{N} L_i(y_i, \hat{y}_i) + \lambda R(\boldsymbol{\theta})$

- 梯度下降: 每次用到所有的训练样本 $\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_{i=1}^{N} \frac{\partial L_i(\boldsymbol{\theta})}{\partial(\boldsymbol{\theta})} + \lambda \frac{\partial R(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$

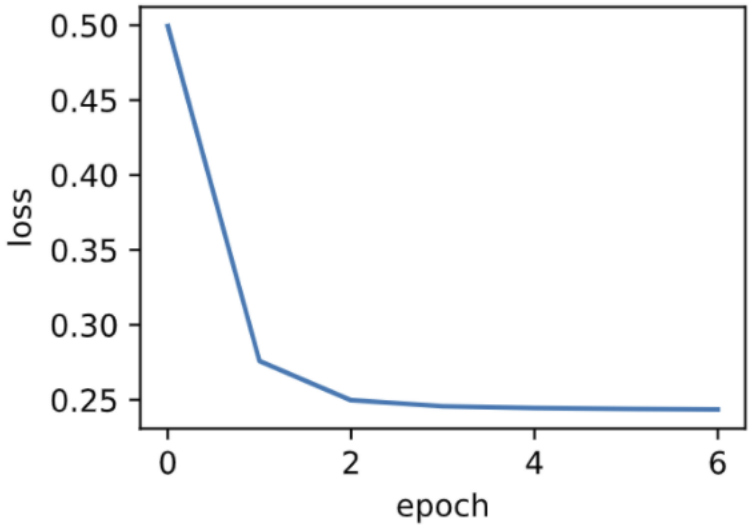
  缺点: 对N个样本求和: 当N很大时, 计算慢

- 随机梯度下降（Stochastic Gradient Descent, SGD）: 每次只计算一个样本上梯度

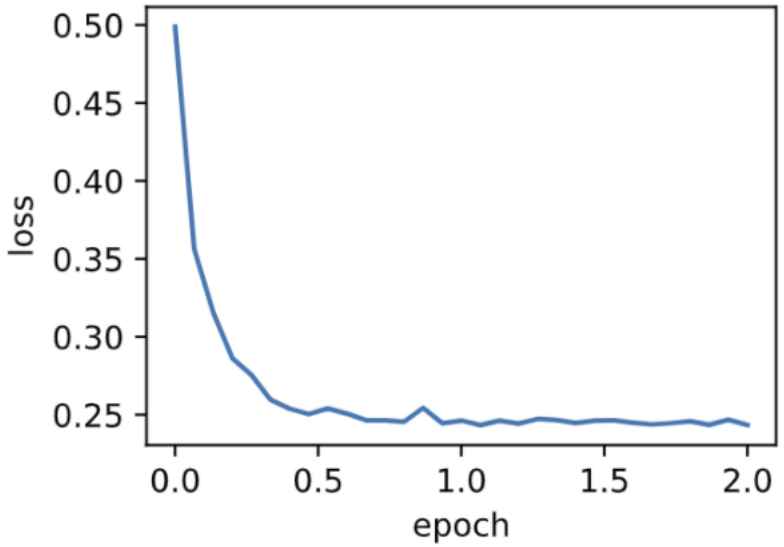
  缺点: 没有有效利用CPU和GPU的计算效率

- 小批量梯度下降（Mini-batch SGD）: 每次只随机送入一定批量的训练样本

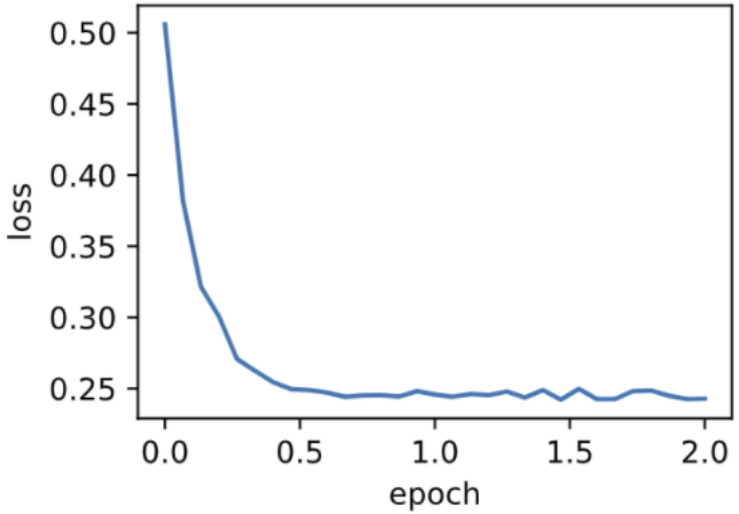
例:



批处理梯度下降
$(bathsize = N = 1500)$



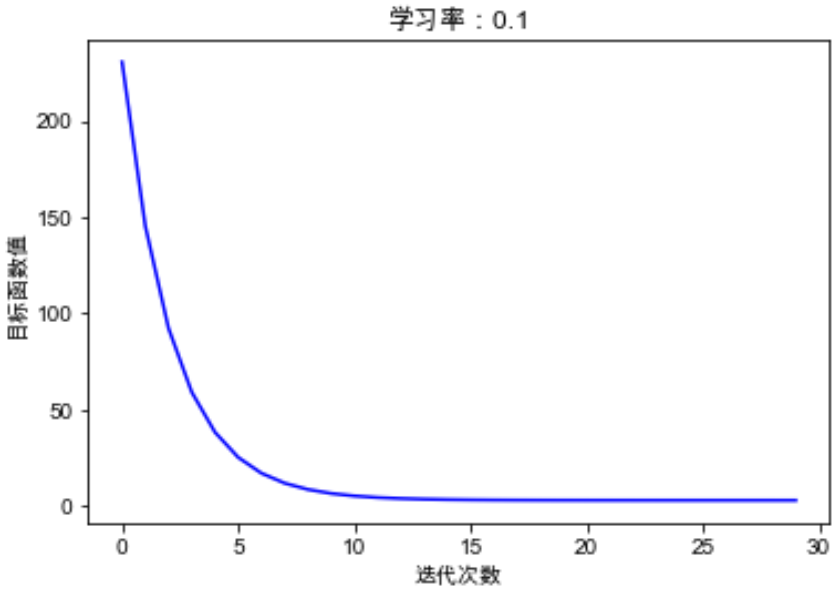
随机梯度下降
$(bathsize = 1)$

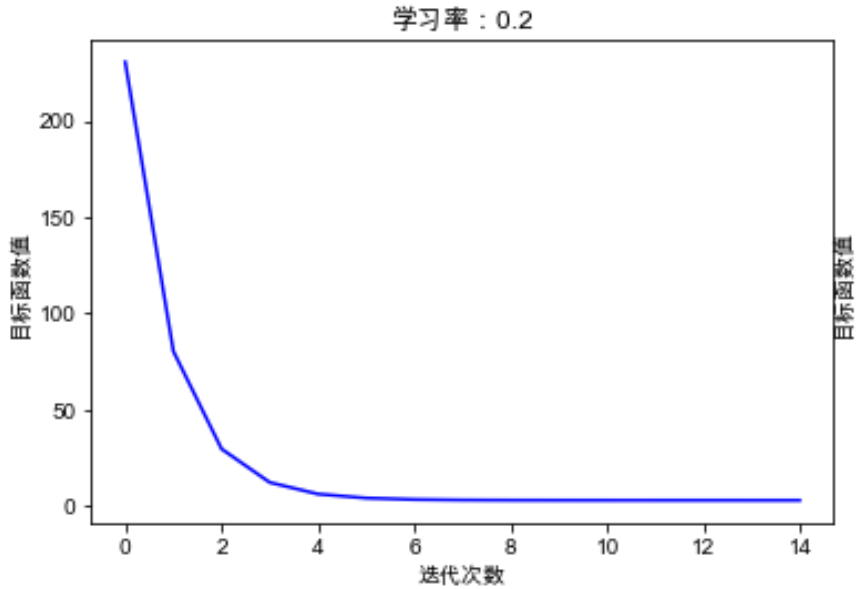


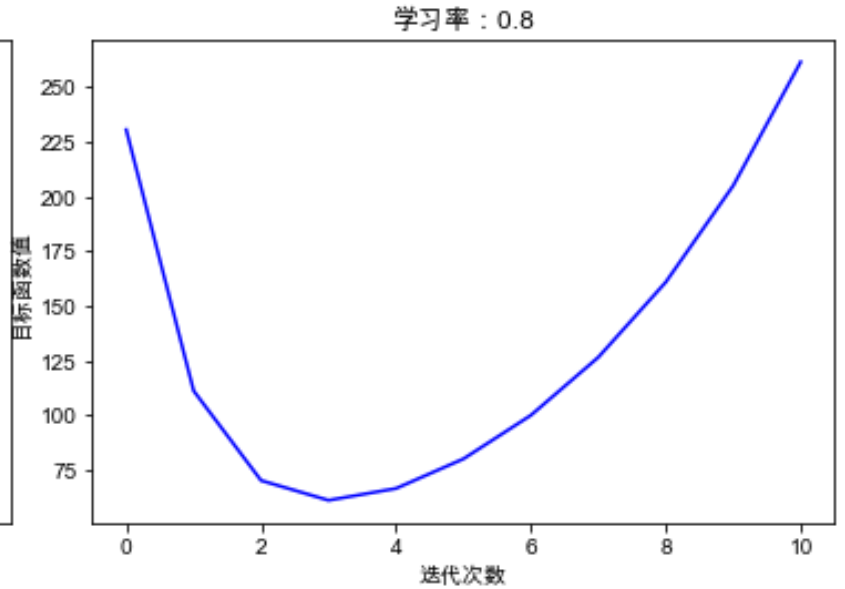
小批量梯度下降
$(bathsize = 10)$

https://tangshusen.me/Dive-into-DL-PyTorch/#/chapter07_optimization/7.3_minibatch-sgd

✓ 学习率







学习率对训练的影响

3.4.6 坐标轴下降法——Lasso求解*

Lasso的目标函数为：$J(\boldsymbol{w}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1$

坐标轴下降法：沿坐标轴方向搜索

- 引入次梯度，在不可微处直接赋值
- 在每次迭代中，在当前点处沿一个坐标轴方向进行一维搜索。
- 循环使用不同的坐标轴。一个周期的一维搜索迭代过程相当于一个梯度迭代。

3.5 模型性能评价

- 均方根误差（Rooted Mean Squared Error，RMSE）

$$\mathrm{RMSE}\,(\boldsymbol{y},\hat{\boldsymbol{y}}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2}$$

- 平均绝对误差（Mean Absolute Error，MAE）

$$\mathrm{MAE}(\boldsymbol{y},\hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{i=1}^{N}|y_i-\hat{y}_i|$$

- R^2分数（R^2score）：既考虑了预测值与真值之间的差异，也考虑了问题本身真值之间的差异

$$SS_{res}\,(\boldsymbol{y},\hat{\boldsymbol{y}}) = \frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2,\ SS_{tot}\,(\boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i-\bar{y})^2,$$

$$R^2(\boldsymbol{y},\hat{\boldsymbol{y}}) = 1-\frac{SS_{res}\,(\boldsymbol{y},\hat{\boldsymbol{y}})}{SS_{tot}(\boldsymbol{y})}$$