# 人工智能之机器学习

## 回归算法

主讲人：李老师

# 课程内容

- 线性回归算法
- 正则化
- Lasso、Ridge

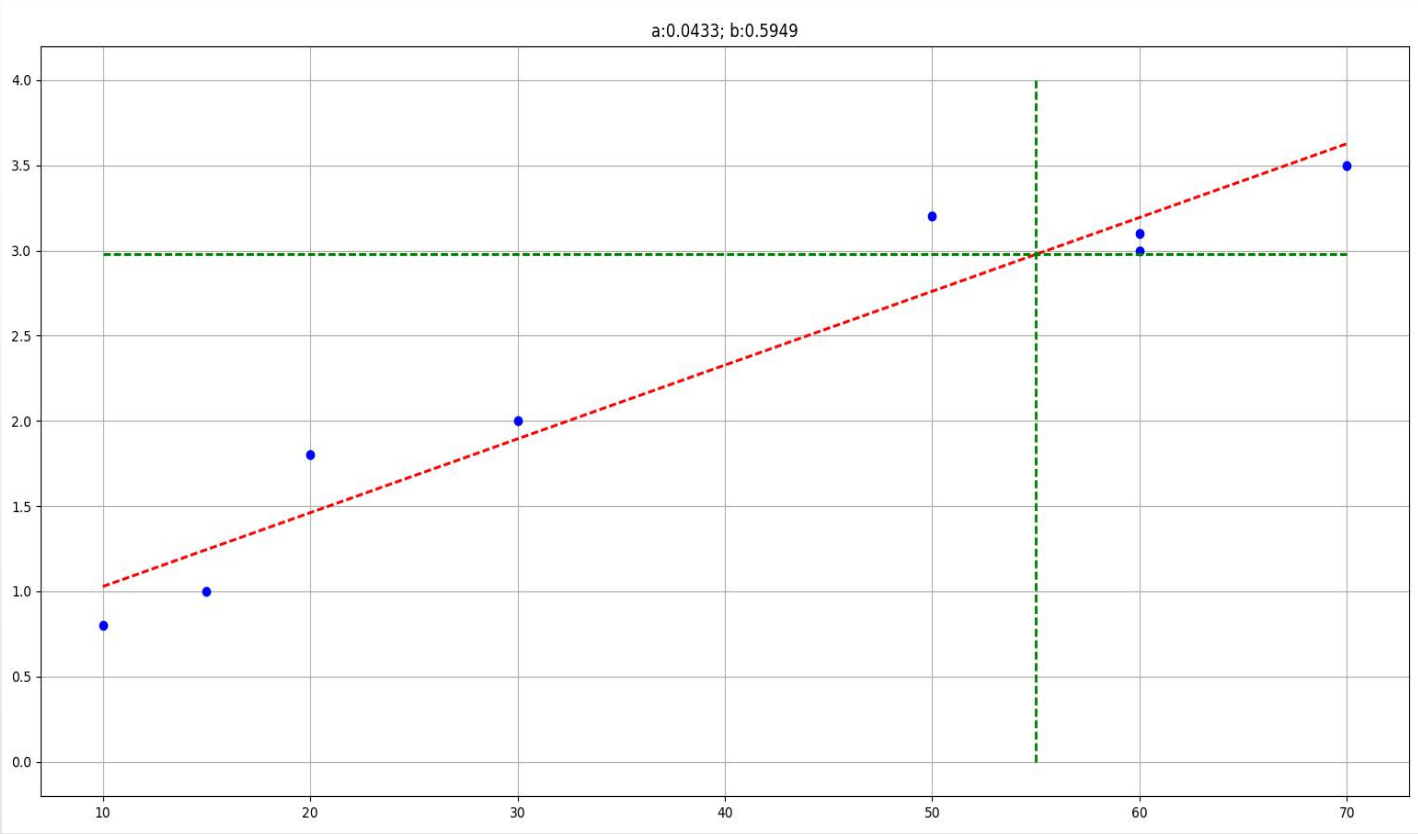
# 什么是回归算法

- 回归算法是一种有监督算法
- 回归算法是一种比较常用的机器学习算法，用来建立“解释”变量(自变量X)和观测值(因变量Y)之间的关系；从机器学习的角度来讲，用于构建一个算法模型(函数)来做属性(X)与标签(Y)之间的映射关系，在算法的学习过程中，试图寻找一个函数 $h: R^d -> R$ 使得参数之间的关系**拟合性**最好。
- 回归算法中算法(函数)的最终结果是一个**连续**的数据值，输入值(属性值)是一个d维度的属性/数值向量

# 线性回归

- y=ax+b

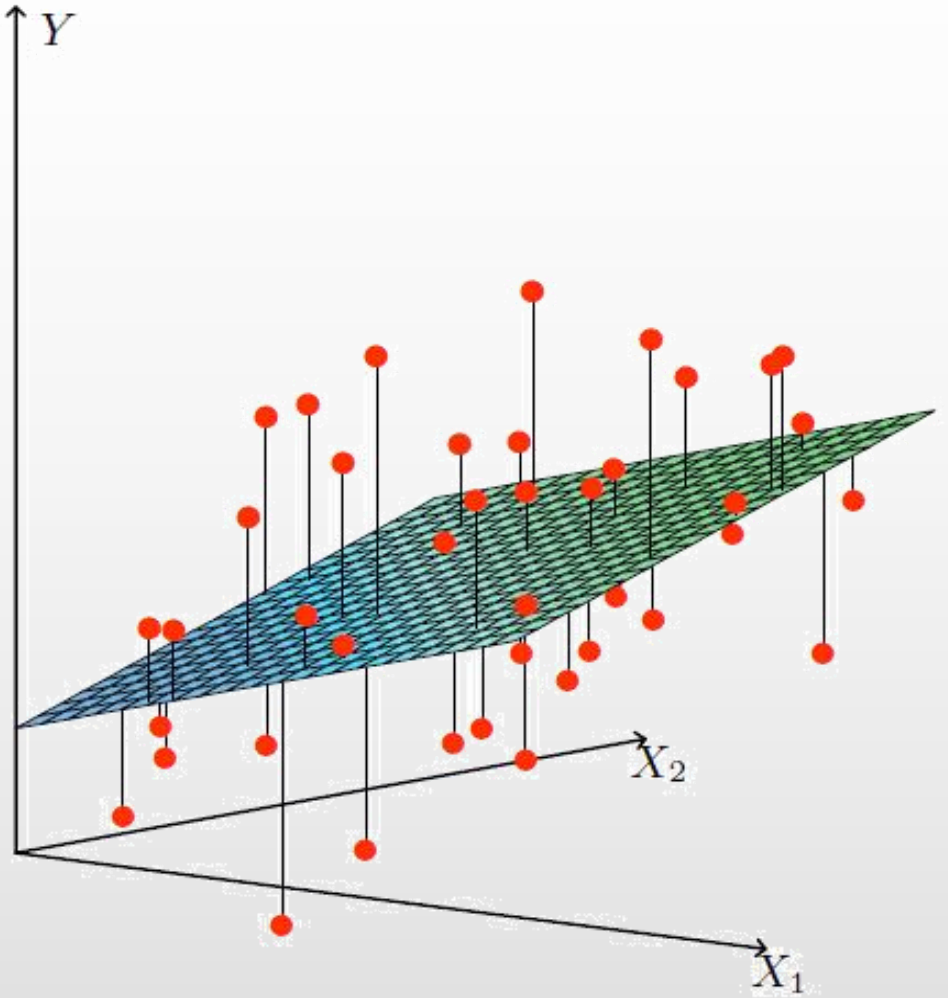
| 房屋面积(m^2) | 租赁价格(1000￥) |
| --- | --- |
| 10 | 0.8 |
| 15 | 1 |
| 20 | 1.8 |
| 30 | 2 |
| 50 | 3.2 |
| 60 | 3 |
| 60 | 3.1 |
| 70 | 3.5 |



# 线性回归

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

| 房屋面积 | 房间数量 | 租赁价格 |
| --- | --- | --- |
| 10 | 1 | 0.8 |
| 15 | 1 | 1.0 |
| 20 | 1 | 1.8 |
| 30 | 1 | 2.0 |
| 50 | 2 | 3.2 |
| 60 | 1 | 3.0 |
| 60 | 2 | 3.1 |
| 70 | 2 | 3.5 |
| ..... | ..... | ..... |



# 线性回归

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

$$= \theta_0 1 + \theta_1 x_1 + \cdots + \theta_n x_n$$

$$= \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

$$= \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

- 最终要求是计算出 $\theta$ 的值，并选择最优的 $\theta$ 值构成算法公式

# 线性回归

- 认为数据中存在线性关系，也就是特征属性X和目标属性Y之间的关系是满足线性关系。
- 在线性回归算法中，找出的模型对象是期望所有训练数据比较均匀的分布在直线或者平面的两侧。
- 在线性回归中，最优模型也就是所有样本(训练数据)离模型的直线或者平面距离最小。

# 最小二乘

- 也就是说我们线性回归模型最优的时候是所有样本的预测值和实际值之间的差值最小化，由于预测值和实际值之间的差值存在正负性，所以要求平方后的值最小化。也就是可以得到如下的一个目标函数：

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(\varepsilon^{(i)}\right)^2=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$$

# 回归算法理性认知

- 房价的预测

| 房屋面积(m^2) | 租赁价格(1000￥) |
| --- | --- |
| 10 | 0.8 |
| 15 | 1 |
| 20 | 1.8 |
| 30 | 2 |
| 50 | 3.2 |
| 60 | 3 |
| 60 | 3.1 |
| 70 | 3.5 |

- 请问，如果现在有一个房屋面积为55平，请问最终的租赁价格是多少比较合适?

# 线性回归、最大似然估计及二乘法

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$$

- 误差 $\varepsilon^{(i)}(1 \le i \le n)$ 是独立同分布的，服从均值为0，方差为某定值 $\sigma^2$ 的**高斯分布**。
  - 原因：**中心极限定理**
- 实际问题中，很多随机现象可以看做**众多因素**的独立影响的综合反应，往往服从正态分布

# 似然函数

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$$

$$p\left(\varepsilon^{(i)}\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{\left(\varepsilon^{(i)}\right)^2}{2\sigma^2}\right)}$$

$$p\left(y^{(i)} \mid x^{(i)}; \theta\right) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

$$L(\theta) = \prod_{i=1}^{m} p\left(y^{(i)} \mid x^{(i)}; \theta\right)$$

$$= \prod_{i=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2}\right)$$

# 对数似然、目标函数及最小二乘

$$\ell(\theta) = \ln L(\theta)$$

$$= \ln \prod_{i=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} \exp\left( -\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2} \right)$$

$$= \sum_{i=1}^{m} \ln \frac{1}{\sigma\sqrt{2\pi}} \exp\left( -\frac{\left(y^{(i)} - \theta^T x^{(i)}\right)^2}{2\sigma^2} \right)$$

$$= m \ln \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{\sigma^2} \bullet \frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \theta^T x^{(i)}\right)^2$$

$$loss(y_j, \hat{y}_j) = J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

# θ的求解过程

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$$

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2=\frac{1}{2}(X\theta-Y)^T(X\theta-Y)\quad\longrightarrow\quad\min_\theta J(\theta)$$

$$\nabla_\theta J(\theta)=\nabla_\theta\left(\frac{1}{2}(X\theta-Y)^T(X\theta-Y)\right)=\nabla_\theta\left(\frac{1}{2}\left(\theta^TX^T-Y^T\right)(X\theta-Y)\right)$$

$$=\nabla_\theta\left(\frac{1}{2}\left(\theta^TX^TX\theta-\theta^TX^TY-Y^TX\theta+Y^TY\right)\right)$$

$$=\frac{1}{2}\left(2X^TX\theta-X^TY-\left(Y^TX\right)^T\right)$$

$$=X^TX\theta-X^TY$$

$$\theta=\left(X^TX\right)^{-1}X^TY$$

# 最小二乘法的参数最优解

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$$

- 参数解析式

$$\theta = \left(X^T X\right)^{-1} X^T Y$$

- 最小二乘法的使用要求矩阵$X^T X$是可逆的；为了防止不可逆或者过拟合的问题存在，可以增加额外数据影响，导致最终的矩阵是可逆的：

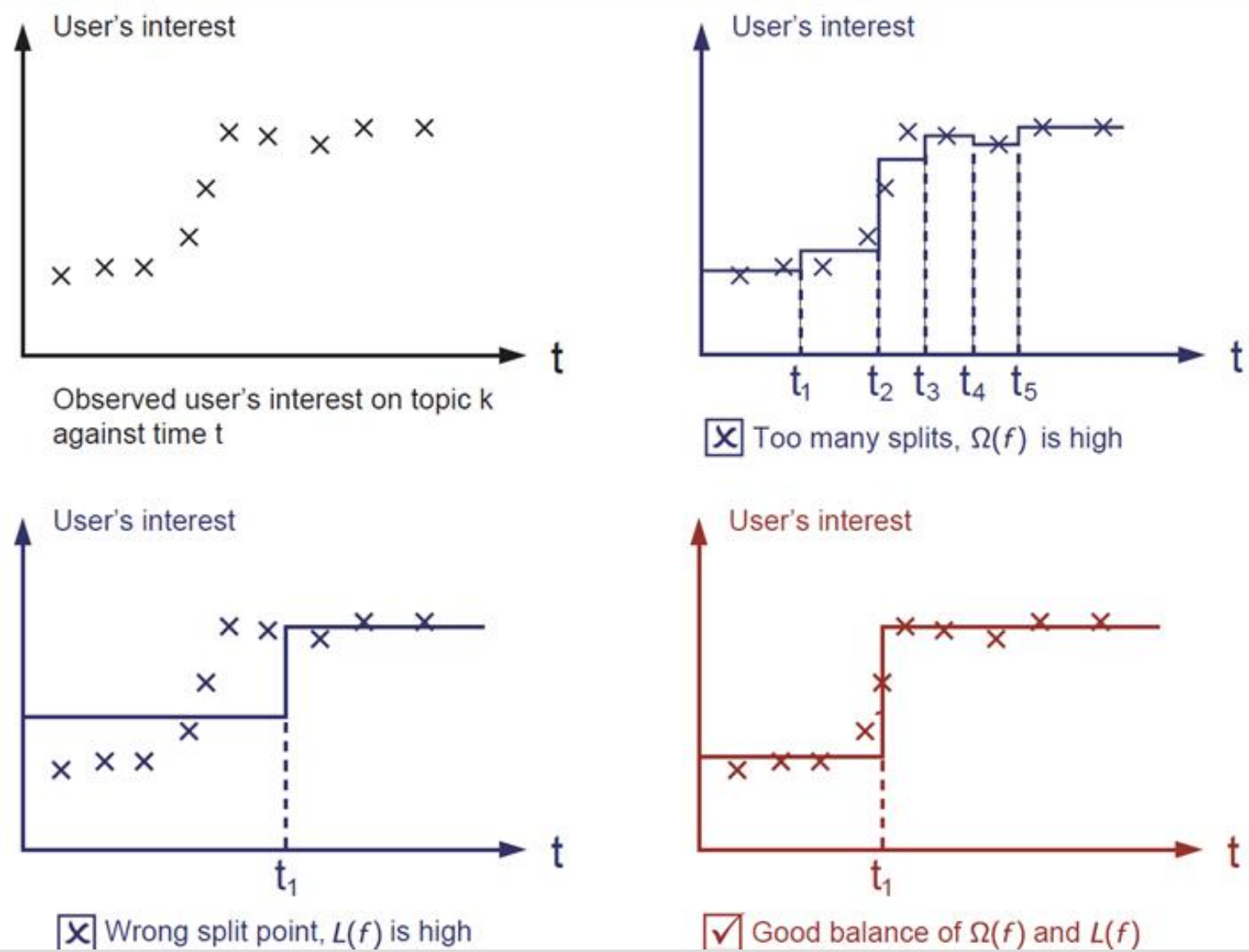
$$\theta = \left(X^T X + \lambda I\right)^{-1} X^T y$$

- 最小二乘法直接求解的难点：矩阵逆的求解是一个难处

# 目标函数(loss/cost function)

- 0-1损失函数 $J(\theta)=\begin{cases}1, Y\neq f(X)\\ 0, Y=f(X)\end{cases}$
- 感知器损失函数 $J(\theta)=\begin{cases}1, |Y-f(X)|>t\\ 0, |Y-f(X)|\leq t\end{cases}$
- 平方和损失函数 $J(\theta)=\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$
- 绝对值损失函数 $J(\theta)=\sum_{i=1}^{m}\left|h_\theta\left(x^{(i)}\right)-y^{(i)}\right|$
- 对数损失函数 $J(\theta)=-\sum_{i=1}^{m}\left(y^{(i)}\log h_\theta\left(x^{(i)}\right)\right)$

# 模型


User's interest
t
Observed user's interest on topic k against time t
User's interest
t
$t_1$ $t_2$ $t_3$ $t_4$ $t_5$
☒ Too many splits, $\Omega(f)$ is high
User's interest
t
$t_1$
☒ Wrong split point, $L(f)$ is high
User's interest
t
$t_1$
☑ Good balance of $\Omega(f)$ and $L(f)$

# 线性回归的过拟合

- 目标函数： $J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2$
- 为了防止数据过拟合，也就是的θ值在样本空间中不能过大，可以在目标函数之上增加一个平方和损失：

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+\lambda\sum_{i=1}^{n}\theta_j^2$$

- **正则项(norm)/惩罚项**： $\lambda\sum_{j=1}^{n}\theta_j^2$ ；这里这个正则项叫做L2-norm

# 过拟合和正则项

- L2-norm:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{i=1}^{n}\theta_j^2 \qquad \lambda > 0$$

- L1-norm:

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\left|\theta_j\right| \qquad \lambda > 0$$

# Ridge回归

- 使用L2正则的线性回归模型就称为Ridge回归(岭回归)

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{i=1}^{n}\theta_j^2 \qquad \lambda > 0$$

# LASSO回归

- 使用L1正则的线性回归模型就称为LASSO回归(Least Absolute Shrinkage and Selection Operator)

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\left|\theta_j\right| \qquad \lambda > 0$$

# Ridge(L2-norm)和LASSO(L1-norm)比较

- L2-norm中，由于对于各个维度的参数缩放是在一个圆内缩放的，不可能导致有维度参数变为0的情况，那么也就不会产生稀疏解；实际应用中，数据的维度中是存在**噪音**和**冗余**的，稀疏的解可以找到有用的维度并且减少冗余，提高后续算法预测的**准确性**和**鲁棒性**（减少了overfitting）(L1-norm可以达到最终解的**稀疏性**的要求)
- Ridge模型具有较高的准确性、鲁棒性以及稳定性(冗余特征已经被删除了)；LASSO模型具有较高的求解速度。
- 如果既要考虑稳定性也考虑求解的速度，就使用Elasitc Net

# Ridge(L2-norm)和LASSO(L1-norm)比较

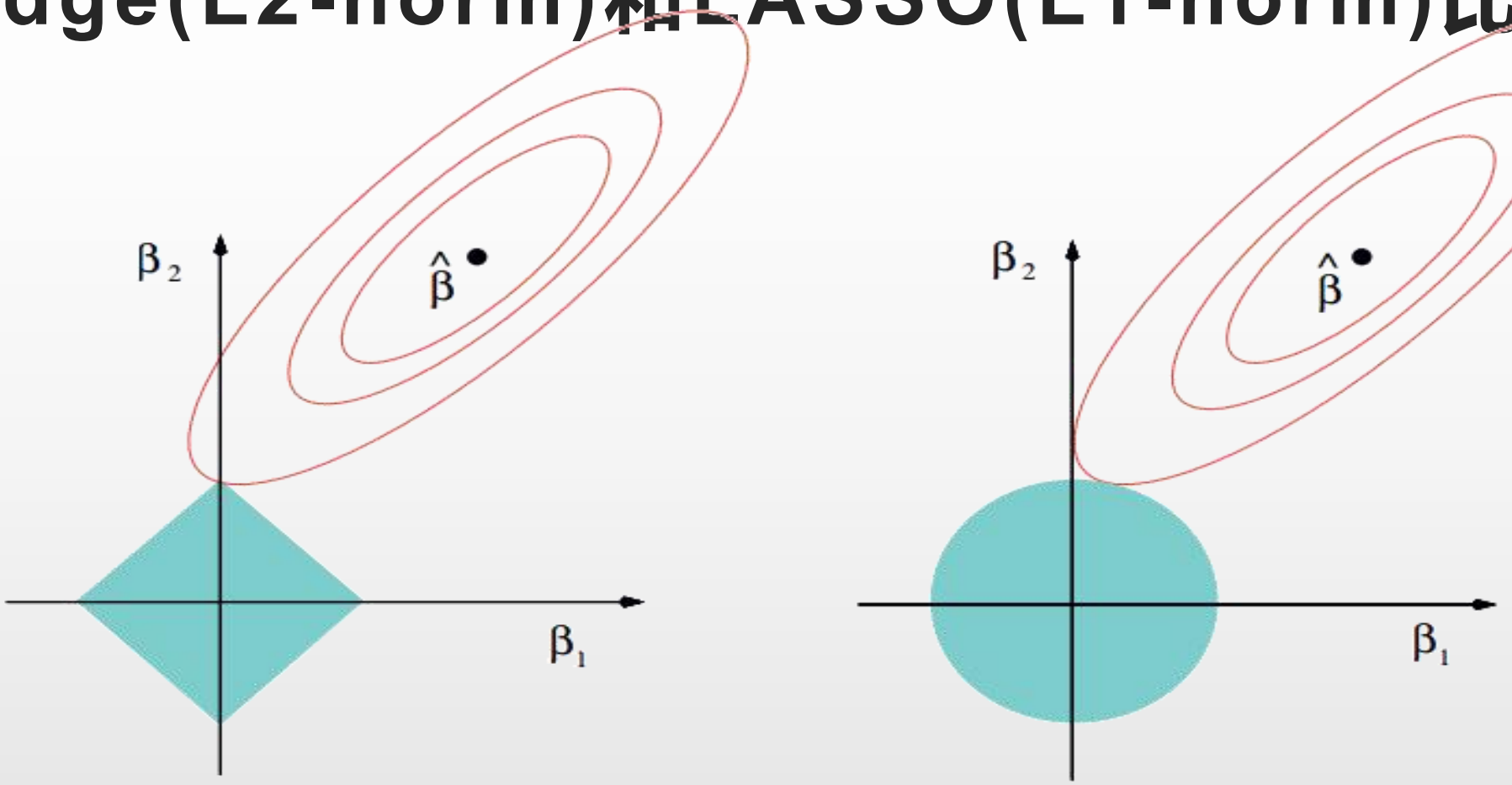



**FIGURE 3.11.** *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

# Elasitc Net

- 同时使用L1正则和L2正则的线性回归模型就称为Elasitc Net算法(弹性网络算法)

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+\lambda\left(p\sum_{j=1}^{n}\left|\theta_j\right|+(1-p)\sum_{j=1}^{n}\theta_j^2\right)$$

$$\begin{cases}\lambda>0\\ p\in[0,1]\end{cases}$$

# 模型效果判断

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2}$$

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{m}\left(y_i - \bar{y}\right)^2} \qquad \bar{y} = \frac{1}{m}\sum_{i=1}^{m} y_i$$

# 模型效果判断

- MSE：误差平方和，越趋近于0表示模型越拟合训练数据。
- RMSE：MSE的平方根，作用同MSE
- $R^2$：取值范围(负无穷,1]，值越大表示模型越拟合训练数据；最优解是1；当模型预测为随机值的时候，有可能为负；若预测值恒为样本期望，$R^2$为0
- TSS：总平方和TSS(Total Sum of Squares)，表示样本之间的差异情况，是伪方差的m倍
- RSS：残差平方和RSS（Residual Sum of Squares），表示预测值和样本值之间的差异情况，是MSE的m倍

# 机器学习调参

- 在实际工作中，对于各种算法模型(线性回归)来讲，我们需要获取θ、λ、p的值；θ的求解其实就是算法模型的求解，一般不需要开发人员参与(算法已经实现)，主要需要求解的是λ和p的值，这个过程就叫做**调参(超参)**
- 交叉验证：将训练数据分为多份，其中一份进行数据验证并获取最优的超参：λ和p；比如：十折交叉验证、五折交叉验证(scikit-learn中默认)等

# 线性回归总结

- 算法模型：线性回归(Linear)、岭回归(Ridge)、LASSO回归、Elastic Net
- 正则化：L1-norm、L2-norm
- 损失函数/目标函数：$J(\theta)=\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2 \longrightarrow \min_\theta J(\theta)$
- θ求解方式：最小二乘法(直接计算，目标函数是平方和损失函数)、梯度下降(BGD\SGD\MBGD)