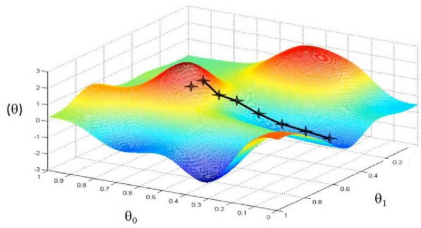
机器学习

计算机学院

杨晓春
xcyang@bit.edu.cn

1

第三章：线性模型



- Tom M. Mitchell, McGraw Hill, 2003
- http://www.cs.cmu.edu/~awm/tutorials
- 周志华，机器学习，2016

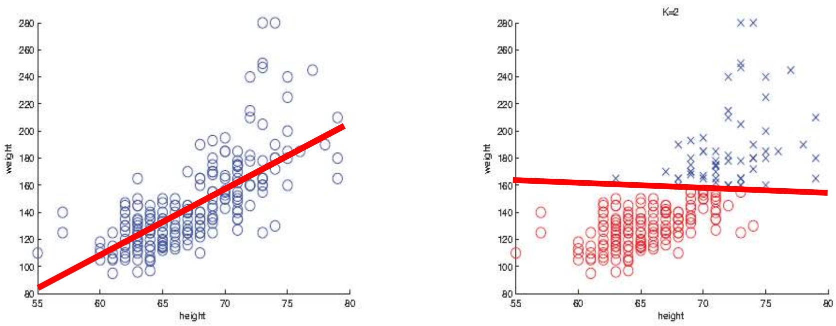
线性模型

- 例子：身高和体重的关系
  - 相关关系，不是确定关系

如何确定相关关系？

- 例子：胖、瘦

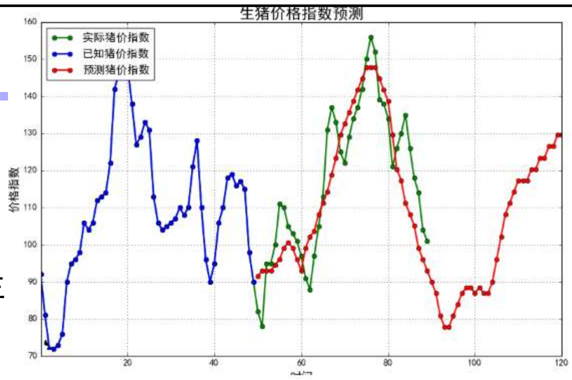
如何进行分类？



(a) (b)

3

线性模型



- 背景
  - 生猪期货
  - 价格波动剧烈
  - 大/中型养殖户为主
  - 直接意义：预测半年到一年后的生猪价格， 对当前养殖规模的确定有重大决策意义。
  - 模型实践：蓝色曲线为历史生猪价格，用于建模；绿色曲线为回测数据，用于验证模型；红色曲线为模型预测结果。

本页课件致谢：普开数据 PUKAI DATA 邹伟

4

大纲

- 线性回归
  - 最小二乘法
  - 梯度下降算法
- 二分类任务
  - 对数几率回归
- 多分类任务
  - 一对一
  - 一对其余
  - 多对多
- 类别不平衡问题

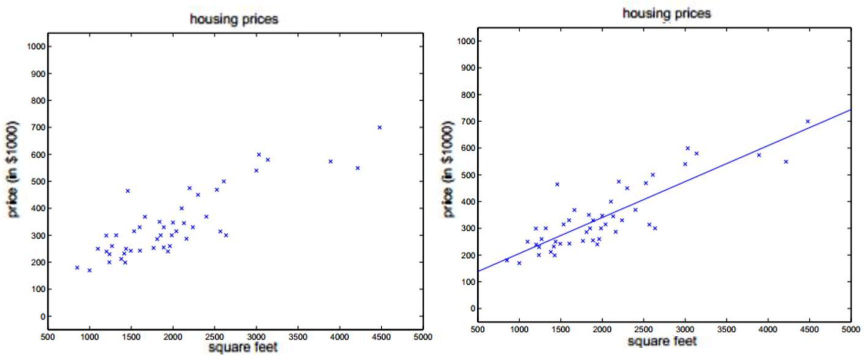
线性回归

- 目的：做预测
- 需要解决2个问题
  - 目标函数
    - 优化问题中的一个概念。任何一个优化问题包括两个部分：(1)目标函数，最终是要最大化或者最小化这个函数；(2)约束条件。约束条件是可选的，比如x<0x<0
  - 损失函数
    - 度量的是预测值与真实值之间的差异

6

单变量形式

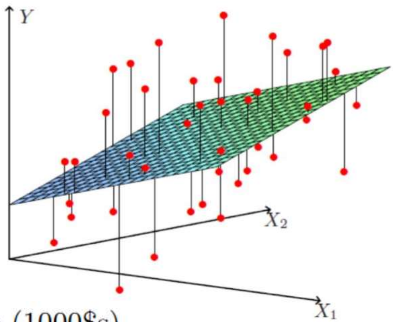
- 线性方程 $y = ax+b$



两个变量的形式

- 线性方程

$$f(\boldsymbol{x}) = w_1x_1 + w_2x_2 + b$$



Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

基本形式

- 线性模型的一般形式

$$f(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \ldots + w_d x_d + b$$

  $\boldsymbol{x} = (x_1; x_2; \ldots; x_d)$ 是由属性描述的列向量，
  其中 x_i 是 $\boldsymbol{x}$ 在第 i 个属性上的取值

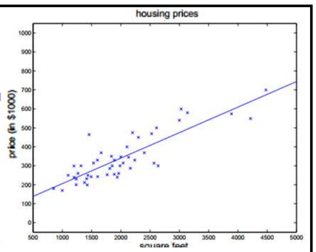
- 向量形式 $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} + b$，其中列向量 $\boldsymbol{w} = (w_1; w_2; \ldots; w_d)$
- 一个例子
  - 综合考虑色泽、根蒂和敲声来判断西瓜好不好
  - 其中根蒂的系数最大，表明根蒂最要紧；而敲声的系数比色泽大，说明敲声比色泽更重要

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

线性模型优点

- 形式简单、易于建模
- 可解释性
- 非线性模型的基础
  - 引入层级结构或高维映射

线性回归



- 给定数据集

$$D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$$

其中

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

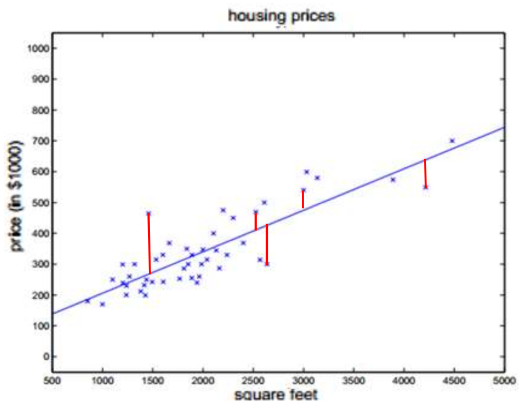
- 线性回归（linear regression）目的
  - 学得一个线性模型以尽可能准确地预测实际输出标记

- 离散属性处理
  - 有“序”关系
    - 连续化为连续值，例如高、中、低 对应 {1.0, 0.5, 0.0}
  - 无“序”关系
    - 有k个属性值，则转换为k维向量
    - 例如西瓜、南瓜、黄瓜 对应 (0,0,1), (0,1,0), (1,0,0)

线性回归

- 单一属性的线性回归
  - $f(x_i)$ 是预测值，y_i是真实值

$$f(x_i)=wx_i+b \ \text{使得}\ f(x_i)\simeq y_i$$

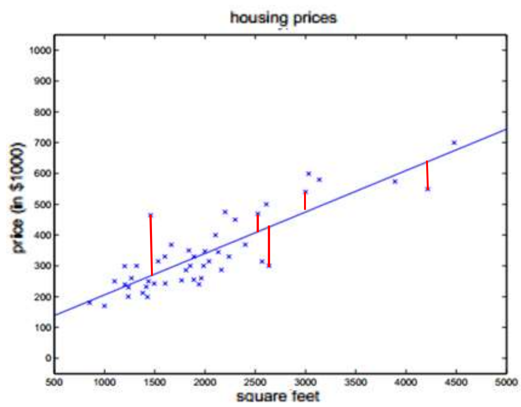


关键：衡量y_i与$f(x_i)$ 间的差别

让差别尽可能小

线性回归 - 最小二乘法（least square method）

- 最小化均方误差
  - 找到一条直线，使所有样本(真实值)到直线(预测值)上的欧式距离之和最小



关键：衡量y_i与$f(x_i)$间的差别

让差别尽可能小

13

线性回归 - 最小二乘法（least square method）

- 单一属性的线性回归目标
  - $f(x_i)$是预测值，y_i是真实值

$$f(x_i) = wx_i + b \text{ 使得 } f(x_i) \simeq y_i$$

- 参数/模型估计：最小二乘法
  - 均方误差最小化 – 均方误差是回归任务最常用的性能度量

$$(w^*, b^*) = \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$

$$= \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} (y_i - wx_i - b)^2$$

本页课件来源：周志华《机器学习》及其课件，致谢：李绍园，刘冲

14

线性回归 - 最小二乘法（least square method）

- 最小化均方误差 $E_{(w,b)}=\sum_{i=1}^{m}(y_i-wx_i-b)^2$
- 目的：求解w 和 b，使 $E_{(w,b)}$ 最小化的过程叫：线性回归
  （注意：x_i 和y_i 是已知的）
- 分别对w和b求导，可得 $\boxed{\frac{\partial E_{(w,b)}}{\partial w}=2\left(w\sum_{i=1}^{m}x_i^2-\sum_{i=1}^{m}(y_i-b)\,x_i\right)}$

推导过程

$$\frac{\partial E_{(w,b)}}{\partial w}=\frac{\partial\sum_{i=1}^{m}((wx_i)^2-2(y_i-b)wx_i+(y_i-b)^2)}{\partial w}=\sum_{i=1}^{m}(2w{x_i}^2-2(y_i-b)x_i)$$

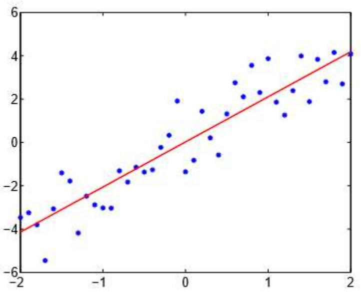
$$\boxed{\frac{\partial E_{(w,b)}}{\partial b}=2\left(mb-\sum_{i=1}^{m}(y_i-wx_i)\right)}$$

15

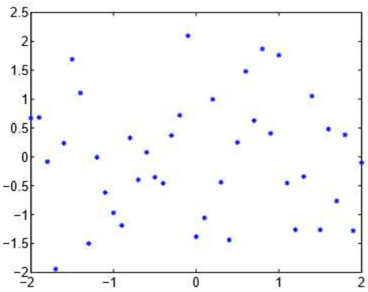
线性回归 - 最小二乘法（least square method）

- 令 $\frac{\partial E_{(w,b)}}{\partial w}=2\left(w\sum_{i=1}^{m}x_i^2-\sum_{i=1}^{m}(y_i-b)\,x_i\right)=0$ $\frac{\partial E_{(w,b)}}{\partial b}=2\left(mb-\sum_{i=1}^{m}(y_i-wx_i)\right)=0$
- 则意味着预测误差(y_i-wx_i-b)均值为0，说明预测误差与输入x_i无关。

$$\sum_{i=1}^{m}(y_i-wx_i-b)x_i=0$$



$$\sum_{i=1}^{m}(y_i-wx_i-b)=0$$



16

线性回归 - 最小二乘法（least square method）

$f(x_i) = wx_i + b$ 使得 $f(x_i) \simeq y_i$

$$\sum_{i=1}^{m}(y_i - wx_i - b) = 0 \qquad \sum_{i=1}^{m}(y_i - wx_i - b)x_i = 0$$

可以写成矩阵的形式

$$\begin{aligned} b\left(\sum_{i=1}^m 1\right) &+ w\left(\sum_{i=1}^m x_i\right) = \sum_{i=1}^m y_i \\ b\left(\sum_{i=1}^m x_i\right) &+ w\left(\sum_{i=1}^m x_i^2\right) = \sum_{i=1}^m y_i x_i \end{aligned}$$

进而写成 $\boldsymbol{w}\Phi = \boldsymbol{t}$，其中 $\boldsymbol{w} = [b \;\; w]$

$$\Phi = \begin{bmatrix} \sum_{i=1}^m 1 & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 \end{bmatrix}, \quad \boldsymbol{t} = \begin{bmatrix} \sum_{i=1}^m y_i \\ \sum_{i=1}^m y_i x_i \end{bmatrix}$$

如果 Φ 可逆，可以得到 $\boldsymbol{w}$ 的参数估计 $\hat{\boldsymbol{w}}$

$$\hat{\boldsymbol{w}} = \Phi^{-1}\boldsymbol{t}$$

线性回归 - 最小二乘法（least square method）

$f(x_i) = wx_i + b$ 使得 $f(x_i) \simeq y_i$

在一个矩阵表示中，需要最小化下面的取值

$$\frac{1}{2}\left\| \begin{bmatrix} y_1 \\ \cdots \\ y_n \end{bmatrix} - \begin{bmatrix} 1 & x_1 \\ \cdots & \cdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \right\|^2 \quad \text{或} \quad \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$$

线性回归的目标函数

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}(f(x_i) - y_i)^2$$

通过将导数设为0，可得

$$\mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{X}\mathbf{w} = 0 \quad \Rightarrow \quad \hat{\mathbf{w}} = \underbrace{(\mathbf{X}^T\mathbf{X})}_{\Phi}{}^{-1}\underbrace{\mathbf{X}^T\mathbf{y}}_{\boldsymbol{t}}$$

该结果是对y的一个线性函数

多元线性回归

- 给定数据集

$$D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$$

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

- 多元线性回归目标

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

多元线性回归

- 把$\boldsymbol{w}$ 和b 吸收入向量形式 $\hat{\boldsymbol{w}}=(\boldsymbol{w};b)$ 数据集表示为

$$\mathbf{X}=\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1\\ x_{21} & x_{22} & \cdots & x_{2d} & 1\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix}=\begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} & 1\\ \boldsymbol{x}_2^{\mathrm{T}} & 1\\ \vdots & \vdots\\ \boldsymbol{x}_m^{\mathrm{T}} & 1 \end{pmatrix}$$

$$\boldsymbol{y}=(y_1;y_2;\ldots;y_m)$$

多元线性回归 - 最小二乘法

- 最小二乘法（least square method）

$$\hat{\boldsymbol{w}}^* = \underset{\hat{\boldsymbol{w}}}{\arg\min} \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)^{\mathrm{T}} \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)$$

令 $E_{\hat{\boldsymbol{w}}} = \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)^{\mathrm{T}} \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)$ ，对 $\hat{\boldsymbol{w}}$ 求导得到

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

$$\begin{aligned}\frac{\partial E_{\hat{w}}}{\partial \hat{w}} &= (y - X\hat{w})\frac{\partial(-X\hat{w})^T}{\partial \hat{w}} + (y^T - (X\hat{w})^T)(-X) \\ &= (y - X\hat{w})(-X^T) + ((X\hat{w})^T - y^T)(X) \\ &= (X\hat{w} - y)(X^T) + ((X\hat{w})^T - y^T)(X) \\ &= (X\hat{w} - y)(X^T) + ((X\hat{w}) - y)(X^T)\end{aligned}$$

$$\frac{\partial E_{\hat{\boldsymbol{w}}}}{\partial \hat{\boldsymbol{w}}} = 2\mathbf{X}^{\mathrm{T}}\left(\mathbf{X}\hat{\boldsymbol{w}} - \boldsymbol{y}\right)$$

令左式为零可得 $\hat{\boldsymbol{w}}$ 最优解的闭式解

21

多元线性回归 - 满秩讨论

- 希望 $\mathbf{X}\hat{\boldsymbol{w}}^* = \boldsymbol{y}$ ➔ $\mathbf{X}^{\mathrm{T}}\mathbf{X}\hat{\boldsymbol{w}}^* = \mathbf{X}^{\mathrm{T}}\boldsymbol{y}$
- 则 $\hat{\boldsymbol{w}}^* = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$

其中 $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}$ 是 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 的逆矩阵，线性回归模型为

$$f\left(\boldsymbol{x}_i\right) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b \quad \Rightarrow \quad f\left(\hat{\boldsymbol{x}}_i\right) = \hat{\boldsymbol{x}}_i^{\mathrm{T}}\underbrace{\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}}_{\hat{\boldsymbol{w}}^*}$$

- 如果 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 不是满秩矩阵？
  - 满秩的含义：在每个维都能观察到方差，否则就是奇异的
  - 不满秩：得到的不是唯一的解
  - 满秩才能保证可逆
  - 解决奇异的方法
    - 根据归纳偏好选择解
    - 引入正则化：$\mathbf{X}^{\mathrm{T}}\mathbf{X} + \alpha I$

22

正则化 及 防止过拟合

线性回归的不同的目标函数 $J(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{m}(f(x_i) - y_i)^2$

- **L1范式**
  LASSO回归

$$J(w) = \underbrace{\frac{1}{2}\sum_{i=1}^{m}(f(x_i) - y_i)^2}_{\text{损失函数}} + \underbrace{\lambda\sum_{j=1}^{d}|w_i|}_{\text{正则项}}$$

很多参数接近0,
说明有些特征没用,
可以降维

- **L2范式**
  Ridge回归

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}(f(x_i) - y_i)^2 + \lambda\sum_{j=1}^{d}{w_i}^2$$

不能降维

- **弹性网Elastic Net**

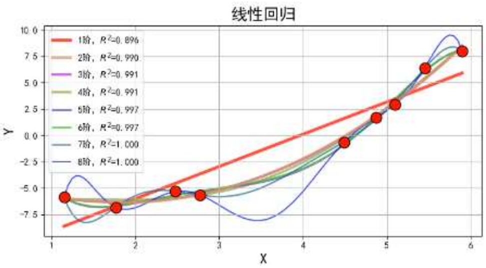
$$\lambda > 0, \rho \in [0, 1]$$

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}(f(x_i) - y_i)^2 + \lambda(\rho\sum_{j=1}^{d}|w_i| + (1-\rho)\sum_{j=1}^{d}{w_i}^2)$$

23

多项式曲线拟合比较（正则化）

- **线性回归**
  - 9个点：$(x_1,y_1), (x_2,y_2),\ldots, (x_9,y_9), (x_1^2,y_1^2), (x_2^2,y_2^2), \ldots, (x_9^2,y_9^2)$
  - 参数：$w_1, w_2,\ldots, w_{18}$
  - 8次方：过拟合



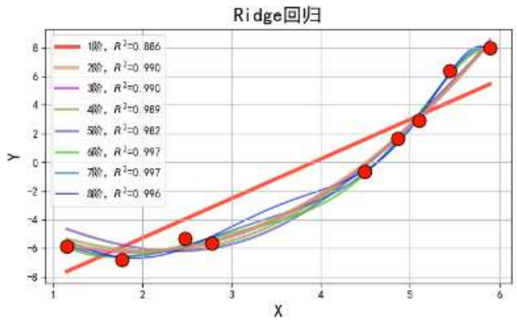

```
线性回归：1阶，系数为：  [-12.12113792   3.05477422]
线性回归：2阶，系数为：  [-3.23812184 -3.36390661  0.90493645]
线性回归：3阶，系数为：  [-3.90207326 -2.61163034  0.66422328  0.022904
线性回归：4阶，系数为：  [-8.20599769  4.20778207 -2.85304163  0.739023
线性回归：5阶，系数为：  [ 21.59733285 -54.12232017  38.43116219 -12.68
线性回归：6阶，系数为：  [ 14.73304784 -37.87317493  23.67462341  -6.07
线性回归：7阶，系数为：  [ 314.30344773 -827.89447316  857.33293588 -46
线性回归：8阶，系数为：  [-1189.50198207  3643.69252986 -4647.93115
```

权重太大，相应的特征依赖太强

24

多项式曲线拟合比较（正则化）

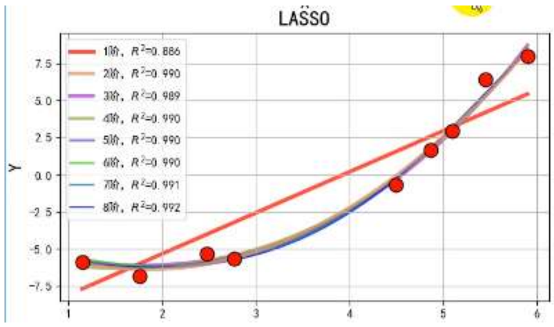
- Ridge回归




```
Ridge回归：1阶，alpha=0.166810，系数为： [-10.79755177   2.75712205]
Ridge回归：2阶，alpha=0.166810，系数为： [-2.86616277 -3.50791358  0.9
Ridge回归：3阶，alpha=0.046416，系数为： [-3.54779374 -2.8374223   0.7
Ridge回归：4阶，alpha=0.166810，系数为： [-3.04995117 -2.03455252 -0.2
Ridge回归：5阶，alpha=0.599484，系数为： [-2.11991122 -1.79172368 -0.8
Ridge回归：6阶，alpha=0.001000，系数为： [ 0.53724068 -6.00552086 -3.7
Ridge回归：7阶，alpha=0.046416，系数为： [-2.3505499  -2.24317832 -1.4
Ridge回归：8阶，alpha=0.166810，系数为： [-2.12001325 -1.87286852 -1.2
```


多项式曲线拟合比较（正则化）

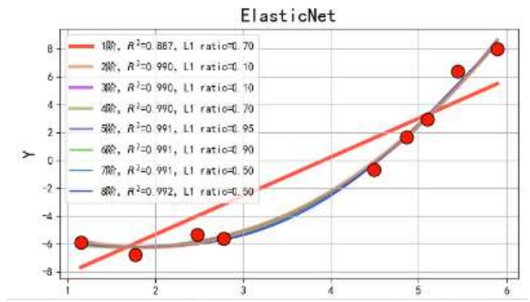
- LASSO回归
  - 高阶模型差不多




```
LASSO：2阶，alpha=0.001000，系数为： [-3.29932625 -3.31989869  0.8987
LASSO：3阶，alpha=0.046416，系数为： [-4.64384442 -1.41251028  0.2190
LASSO：4阶，alpha=0.001000，系数为： [-5.10441283 -1.40548737  0.3404
LASSO：5阶，alpha=0.046416，系数为： [-4.06779799 -1.87958288  0.2539
LASSO：6阶，alpha=0.046416，系数为： [-3.79737378 -1.94437059  0.1965
LASSO：7阶，alpha=0.001000，系数为： [-4.51456835 -1.58477275  0.2348
LASSO：8阶，alpha=0.001000，系数为： [-4.62623251 -1.37717809  0.1718
 0.00629505  0.00069171  0.0000355  -0.00000875 -0.00000386]
```

多项式曲线拟合比较（正则化）

- ElasticNet 弹性网



27

机器学习与数据使用

训练数据 → *w*		

训练数据 → *w*	测试数据

训练数据 → *w*	验证数据 → λ	测试数据
	超参数	

- 交叉验证
  - 例如：十折交叉

本页课件致谢：普开数据 PUKAI DATA 邹伟

28

大纲

- 线性回归
  - 最小二乘法
  - 梯度下降算法
- 二分类任务
  - 对数几率回归
  - 线性判别分析
- 多分类任务
  - 一对一
  - 一对其余
  - 多对多
- 类别不平衡问题

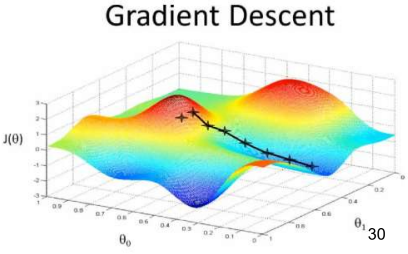
29

梯度下降算法

- 目标函数 $J(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{m}(f(x_i) - y_i)^2$
- 初始化$\boldsymbol{w}$ (随机初始化)
- 沿着负梯度方向迭代，更新后的$\boldsymbol{w}$使$J(\boldsymbol{w})$更小

$$\boldsymbol{w} = \boldsymbol{w} - \alpha \cdot \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}$$

- α：学习率、步长



30

梯度方向

$$\frac{\partial}{\partial w_j} J(\boldsymbol{w}) = \frac{\partial}{\partial w_j} \frac{1}{2} (f_w(\boldsymbol{x}) - y)^2$$

$$= 2 \cdot \frac{1}{2} (f_w(\boldsymbol{x}) - y) \frac{\partial}{\partial w_j} (f_w(\boldsymbol{x}) - y)$$

$$= (f_w(\boldsymbol{x}) - y) \frac{\partial}{\partial w_j} (\sum_0^d w_i x_i - y)$$

$$= (f_w(\boldsymbol{x}) - y) x_i$$

31

批量梯度下降算法

Repeat until convergence {

$$w_j := w_j + \alpha \sum_{i=1}^{m} (y_i - f_w(x_i)) x_{i,j}$$

}

- 通常梯度下降受局部最优影响，但用于线性回归的最优化问题只有唯一的全局最优解，没有其他的局部最优解。因此，总能保证梯度下降算法收敛到全局最小（假设学习率α不是很大）
- $J(\boldsymbol{w})$是个凸二次函数
- 缺点
  - 每做一步，要扫描整个训练集。但m较大时，操作代价大

32

随机(stochastic)梯度下降算法

Loop {
  for i=1 to m {
    $w_j := w_j + \alpha(y_i - f_w(x_i))x_{i,j}$
  }
}

- 也叫增量梯度下降算法
  - 不需一次扫描整个训练集。针对每个样本，都可以改进w的估计。
  - 优点：更快地接近最优的w值的附近
  - 缺点：$J(\boldsymbol{w})$可能永远都不会收敛到最小,而参数$\boldsymbol{w}$在$J(\boldsymbol{w})$的最小值附近振动。实际应用中，大部分在最小附近振动的值已经足够好了。
  - ➔当训练集大时，建议使用该方法

33

折中的方法：mini-batch

Repeat until convergence {
  $$w_j := w_j + \alpha \sum_{i=1}^{m} (y_i - f_w(x_i))x_{i,j}$$
}

Loop {
  for i=1 to m {
    $w_j := w_j + \alpha(y_i - f_w(x_i))x_{i,j}$
  }
}

- mini-batch梯度下降算法
  - 不是每拿到一个样本即更改梯度，而是若干个样本的平均梯度作为更新方向

线性回归中：$f_w(\boldsymbol{x})$是模型的预测值 $f_w(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$

34

大纲

- 线性回归
  - 最小二乘法
  - 梯度下降算法
- 二分类任务
  - 对数几率回归
- 多分类任务
  - 一对一
  - 一对其余
  - 多对多
- 类别不平衡问题

35

对数线性回归

- 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，往往近似服从正态分布
  - 城市耗电量：大量用户的耗电量总和
  - 测量误差：许多观察不到的、微小误差的总和

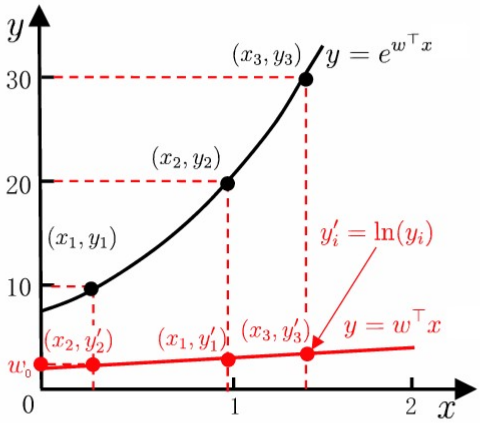
  - 注：应用前提是多个随机变量的和，有些问题是乘性误差，则需要鉴别或者取对数后再使用

36

对数线性回归

- 输出标记的对数为线性模型逼近的目标

将线性模型的预测值和真实标记联系起来



$$\ln y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$$

非线性函数映射

$$y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$$

37

线性回归 - 广义线性模型

- 一般形式

$$y = g\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}\right)$$

- $g(\cdot)$称为联系函数（link function）
  - 单调可微函数

- 对数线性回归是 $g(\cdot) = \ln(\cdot)$ 时广义线性模型的特例

38

二分类任务

- 预测值z与输出标记y

$$z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \qquad y \in \{0,1\}$$

- 寻找函数将分类标记与线性回归模型输出联系起来

- 最理想的函数——单位阶跃函数

$$y = \begin{cases} 0, & z < 0; \\ 0.5, & z = 0; \\ 1, & z > 0, \end{cases}$$

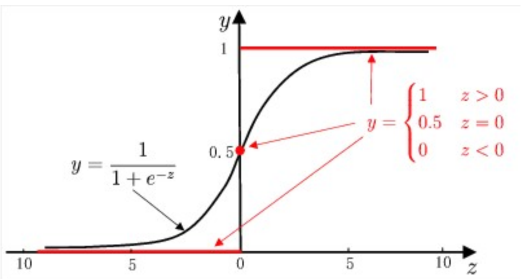
  – 预测值大于零就判为正例，小于零就判为反例，预测值为临界值零则可任意判别

二分类任务

- 单位阶跃函数缺点
  – 不连续
- 替代函数——对数几率函数 (logistic/sigmoid function)
  – 单调可微、任意阶可导

单位阶跃函数与对数几率函数的比较

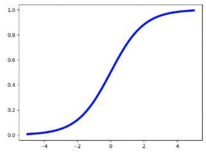
$$y = \frac{1}{1+e^{-z}}$$



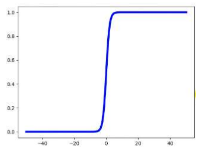
对数几率回归（logistic regression）

- 又称为：逻辑回归
- Logistic/Sigmoid函数

$$f_w(\boldsymbol{x}) = g(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}}$$



x \in [-5,5],
100个采样点



x \in [-50,50],
100个采样点

- 推导

$$g(z) = \frac{1}{1+e^{-z}} \Rightarrow$$

$$g(z)' = \left(\frac{1}{1+e^{-z}}\right)' = \frac{e^{-z}}{(1+e^{-z})^2}$$

$$= \frac{1}{1+e^{-z}} \cdot \frac{e^{-z}}{1+e^{-z}}$$

$$= \frac{1}{1+e^{-z}} \cdot \left(1 - \frac{1}{1+e^{-z}}\right)$$

$$= g(z) \cdot (1 - g(z))$$

41

对数几率回归 -- 参数估计

- 假定

$$p\,(y=1 \mid \boldsymbol{x};\boldsymbol{w}) = f_w(\boldsymbol{x})$$

$$p\,(y=0 \mid \boldsymbol{x};\boldsymbol{w}) = 1 - f_w(\boldsymbol{x})$$

显然有

$$p(y|\boldsymbol{x};\boldsymbol{w}) = (f_w(\boldsymbol{x}))^y(1-f_w(\boldsymbol{x}))^{1-y}$$

Logistic回归的参数估计

$$L(\boldsymbol{w}) = \Pi_{i=1}^m\, p(y_i|x_i,w_i) = \Pi_{i=1}^m\,(f_w(x_i))^{y_i}(1-f_w(x_i))^{1-y_i}$$

42

对数似然函数

$$l(\boldsymbol{w}) = \log L(\boldsymbol{w}) = \sum_{i=1}^{m}(y_i \log f_w(x_i) + (1-y_i)\log(1-f_w(x_i)))$$

求导

$$\begin{aligned}
\frac{\partial l(\boldsymbol{w})}{\partial w_j} &= \sum_{i=1}^{m}\left(\frac{y_i}{f_w(x_i)} - \frac{1-y_i}{1-f_w(x_i)}\right)\frac{\partial f_w(x_i)}{\partial w_j} \\
&= \sum_{i=1}^{m}\left(\frac{y_i}{g(\boldsymbol{w}^T x_i)} - \frac{1-y_i}{1-g(\boldsymbol{w}^T x_i)}\right)\frac{\partial g(\boldsymbol{w}^T x_i)}{\partial w_j} \\
&= \sum_{i=1}^{m}\left(\frac{y_i}{g(\boldsymbol{w}^T x_i)} - \frac{1-y_i}{1-g(\boldsymbol{w}^T x_i)}\right)\cdot g(\boldsymbol{w}^T x_i)\cdot(1-g(\boldsymbol{w}^T x_i))\frac{\partial \boldsymbol{w}^T x_i}{\partial w_j} \\
&= \sum_{i=1}^{m}(y_i(1-g(\boldsymbol{w}^T x_i)) - (1-y_i)g(\boldsymbol{w}^T x_i))\cdot x_{i,j} \\
&= \sum_{i=1}^{m}(y_i - g(\boldsymbol{w}^T x_i))\cdot x_{i,j}
\end{aligned}$$

32

参数迭代

- Logistic回归参数的学习规则

$$w_j := w_j + \alpha(y_i - f_w(x_i))x_{i,j}$$

Repeat until convergence {
$$w_j := w_j + \alpha(y_i - f_w(x_i))x_{i,j}$$
}

Logistic回归中： $f_w(\boldsymbol{x}) = \dfrac{1}{1+e^{-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}}$

- 与线性回归的结论具有相同的表示形式，但实际不同

Repeat until convergence {
$$w_j := w_j + \alpha\sum_{i=1}^{m}(y_i - f_w(x_i))x_{i,j}$$
}

Loop {
for i=1 to m {
$$w_j := w_j + \alpha(y_i - f_w(x_i))x_{i,j}$$
}
}

线性回归中：$f_w(\boldsymbol{x})$是模型的预测值 $f_w(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$

33

区别与联系

- 区别
  - Logistic回归中（分类）：$f_w(\boldsymbol{x}) = \dfrac{1}{1+e^{-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}}}$
  - 线性回归中（模型的预测值）：$f_w(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$
- 联系
  - 线性回归认为误差是高斯分布、逻辑回归服从两点分布
  - 指数族分布：能够写成指数函数形式的
    - 高斯分布、两点分布、二项分布、泊松分布
    - 当$f(x)>0$，$\exp(\ln f(x)) = f(x)$
  - 用法
    - 做连续值的预测：用线性回归
    - 做样本是离散的分类问题：用Logistic回归
    - 预测跟次数相关的问题：用泊松分布
      - e.g. 区域中犯罪率的个数、细胞培养皿中细胞的个数、单位时间服务的次数
    - 用最大似然估计的套路带进去，可以做各种推广。关键是怎么样建立分布。

45

对数线性模型

- 一个事件的几率odds，是指该事件发生的概率与该事件不发生的概率的比值
- 对数几率回归优点 (是一种分类学习方法)
  - 无需事先假设数据分布
  - 可得到“类别”的近似概率预测
  - 任意阶可导的凸函数➔可直接应用现有数值优化算法求取最优解
- 对数几率：logit函数

$$p(y=1 \mid \boldsymbol{x};\boldsymbol{w}) = f_w(\boldsymbol{x})$$
$$p(y=0 \mid \boldsymbol{x};\boldsymbol{w}) = 1 - f_w(\boldsymbol{x})$$

$$\begin{aligned} logit(p) &= \log\frac{p}{1-p} = \log\frac{f_w(x)}{(1-f_w(x))} \\ &= \log\frac{\frac{1}{1+e^{-w^T x}}}{\frac{e^{-w^T x}}{1+e^{-w^T x}}} \\ &= \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} \end{aligned}$$

46

逻辑回归的损失函数 $y_i \in \{0,1\}$

$$L(w) = \Pi_{i=1}^{m} p_i^{y_i}(1-p_i)^{1-y_i}$$

$$\Rightarrow l(w) = \ln L(w) = \sum_{i=1}^{m} \ln(p_i^{y_i}(1-p_i)^{1-y_i})$$

$$\hat{y}_i = \begin{cases} p_i & y_i = 1 \\ 1-p_i & y_i = 0 \end{cases}$$

因为 $p_i = \dfrac{1}{1+e^{-f(x_i)}}$

所以 $l(w) = \sum_{i=1}^{m} \ln((\dfrac{1}{1+e^{-f(x_i)}})^{y_i}(1-\dfrac{1}{1+e^{f(x_i)}})^{1-y_i})$

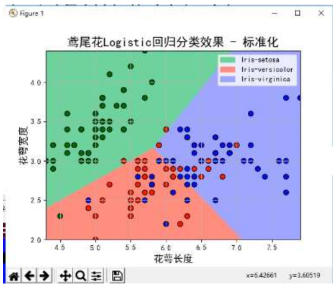
则 $loss(y_i, \hat{y}_i) = -l(w)$

$$= \sum_{i=1}^{m}(y_i \ln(1+e^{-f(x_i)}) + (1-y_i)\ln(1+e^{f(x_i)}))$$

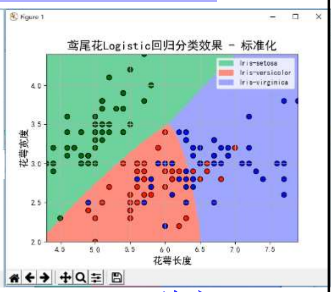
47

对Logistic回归的理解

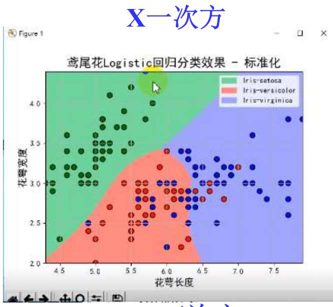
- 线性模型
  - 参数是一阶
  - X可以是高阶
- 叫回归，但是
  - 做分类问题
  - 二分类问题



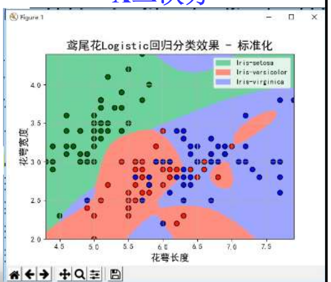
X一次方



X二次方



X三次方



X十次方

48

大纲

- 线性回归
  - 最小二乘法
  - 梯度下降算法
- 二分类任务
  - 对数几率回归
- 多分类学习
  - 一对一
  - 一对其余
  - 多对多
- 类别不平衡问题

多分类学习

- 多分类学习方法
  - 二分类学习方法推广到多类
  - 利用二分类学习器解决多分类问题（常用）
    - 对问题进行拆分，为拆出的每个二分类任务训练一个分类器
    - 对于每个分类器的预测结果进行集成以获得最终的多分类结果
- 拆分策略
  - 一对一（One vs. One, OvO）
  - 一对其余（One vs. Rest, OvR）
  - 多对多（Many vs. Many, MvM）

本页课件来源：周志华《机器学习》及其课件，致谢：李绍园，刘冲

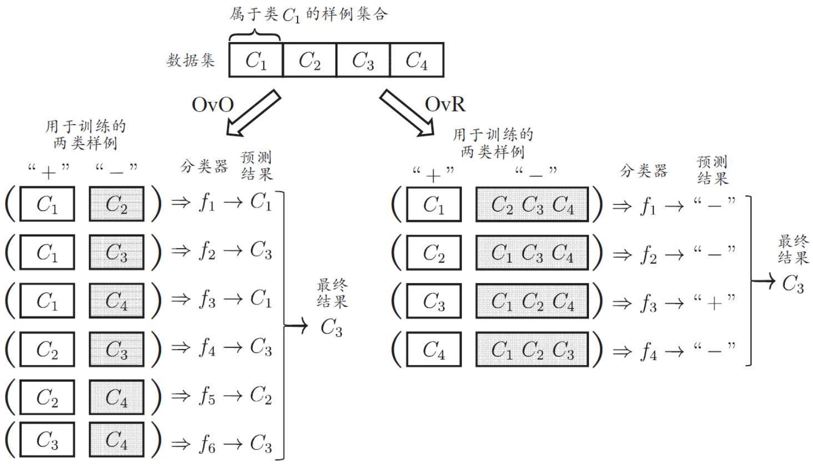
多分类学习– 一对一

- 拆分阶段
  - N个类别两两配对
    - N(N-1)/2 个二类任务
  - 各个二类任务学习分类器
    - N(N-1)/2 个二类分类器
- 测试阶段
  - 新样本提交给所有分类器预测
    - N(N-1)/2 个分类结果
  - 投票产生最终分类结果
    - 被预测最多的类别为最终类别

多分类学习– 一对其余

- 任务拆分
  - 某一类作为正例，其他反例
    - N 个二类任务
  - 各个二类任务学习分类器
    - N 个二类分类器
- 测试阶段
  - 新样本提交给所有分类器预测
    - N 个分类结果
  - 比较各分类器预测置信度
    - 置信度最大类别作为最终类别

多分类学习– 两种策略比较



多分类学习– 两种策略比较

一对一

- 训练N(N-1)/2个分类器，存储开销和测试时间大
- 训练只用两个类的样例，训练时间短

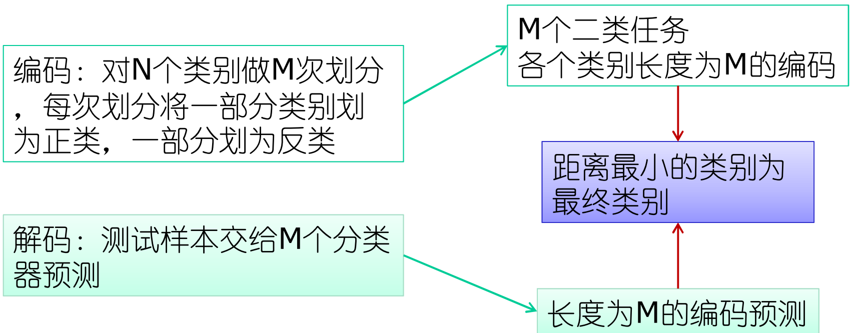
一对其余

- 训练N个分类器，存储开销和测试时间小
- 训练用到全部训练样例，训练时间长

预测性能取决于具体数据分布，多数情况下两者差不多

多分类学习– 多对多

- 多对多（Many vs Many, MvM）
  - 若干类作为正类，若干类作为反类
- 纠错输出码（Error Correcting Output Code, ECOC）



多分类学习– 多对多

- 纠错输出码(Error Correcting Output Code, ECOC)

	f_1	f_2	f_3	f_4	f_5	海明距离	欧氏距离
C_1	−1	+1	−1	+1	+1	3	$2\sqrt{3}$
C_2	+1	−1	−1	+1	−1	4	4
C_3	−1	+1	+1	−1	+1	1	2
C_4	−1	−1	+1	+1	−1	2	$2\sqrt{2}$
测试示例	−1	−1	+1	−1	+1		

(a) 二元 ECOC 码

[Dietterich and Bakiri,1995]

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	海明距离	欧氏距离
C_1	−1	−1	+1	+1	−1	+1	+1	4	4
C_2	−1	0	0	0	+1	−1	0	2	2
C_3	+1	+1	−1	−1	−1	+1	−1	5	$2\sqrt{5}$
C_4	−1	+1	0	+1	−1	0	+1	3	$\sqrt{10}$
测试示例	−1	+1	+1	−1	+1	−1	+1		

0：停用类

(b) 三元 ECOC 码

[Allwein et al. 2000]

- ECOC编码对分类器错误有一定容忍和修正能力，编码越长、纠错能力越强
- 编码长，所需训练的分类器多，需要的存储开销大
- 有限类别，可能的组合数有限，编码长度有上限，太长没有意义
- 对同等长度的编码，理论上来说，任意两个类别之间的编码距离越远，则纠错能力越强

大纲

- 线性回归
  - 最小二乘法
  - 梯度下降算法
- 二分类任务
  - 对数几率回归
- 多分类学习
  - 一对一
  - 一对其余
  - 多对多
- 类别不平衡问题

57

类别不平衡问题

- 类别不平衡（class imbalance）
  - 不同类别训练样例数相差很大情况（正类为小类）

做决策的依据　　　　实际执行的依据

类别平衡正例预测 $\frac{y}{1-y} > 1$ 　　$\frac{y}{1-y} > \frac{m^+}{m^-}$ 正负类比例

反映正例可能性与反例可能性的比值　　观测的真实几率

- 再缩放（基本策略）

$$\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+}$$

实际操作困难：因为假设“训练集是真实样本总体的无偏采样”不成立
因此未必能通过训练样例推断出真实的比例关系

  - 欠采样（undersampling）
    - 去除一些反例使正反例数目接近（EasyEnsemble [Liu et al.,2009]）
  - 过采样（oversampling）
    - 增加一些正例使正反例数目接近（SMOTE [Chawla et al.2002]）
  - 阈值移动（threshold-moving）

58

结论

- 各任务下（回归、分类）各个模型优化的目标
  - 最小二乘法：最小化均方误差
  - 对数几率回归：最大化样本分布似然

本页课件来源：周志华《机器学习》及其课件，致谢：李绍园，刘冲

实验环境

- Pycharm/vs code
- python版本采用anaconda自带的python3.7
- 第三方库：sklearn，graphviz
- 测试一下graphviz是否导入成功