CSDN
不 止 于 代 码

第三讲 极大似然估计

CSDN学院

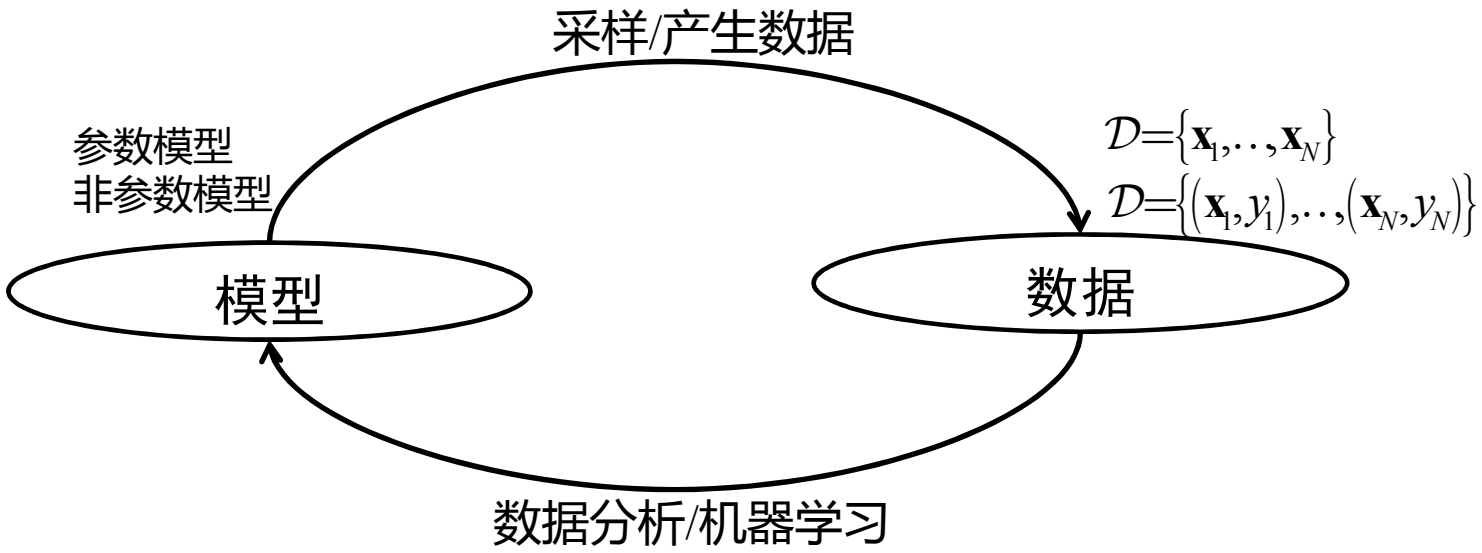
2017年7月

AI 101

统计推断

- 前两讲我们学习了一些概率模型
  - Bernoulli分布、正态分布…
- 接下来两讲，我们来学习统计推断：数据→分布的性质
  - 今天：极大似然估计
  - 明天：贝叶斯估计

概率模型和数据



机器学习：怎样将模型拟合数据？模型拟合.
怎么选择最适合数据的模型？ 模型选择.

IID样本

- 当 $X_1,...,X_N$ 互相独立且有相同的分布F时，记为$X_1,...,X_N \sim F$ ，我们称$X_1,...,X_N$ 为独立同分布（ Independent Identically Distribution, IID）样本，表示 $X_1,...,X_N$ 是从相同分布独立抽样/采样，我们也称 $X_1,...,X_N$是分布F的随机样本。
  - 统计推断中通常假设数据都是来自相同分布的IID样本

参数估计

- 给定模型类别$p(\mathbf{x}|\boldsymbol{\theta})$ 和数据$\mathcal{D}$，选择与数据最匹配的参数$\boldsymbol{\theta}$：参数估计

- 有多种方法可用来估计模型的参数
  - 矩估计法
  - 极大似然估计：频率学派
  - 贝叶斯方法：贝叶斯学派

Outline

- 极大似然估计的基本思想
  - 似然函数、log似然
  - 最大似然 vs. 最小损失
- 常见分布的参数的极大似然估计
  - 正态分布、Bernoulli分布、 Binomial 分布、Multinomial分布
- 一些机器学习模型的参数估计
  - 线性回归、Logistic回归、朴素贝叶斯
- 估计的评价
  - 偏差、方差、偏差-方差分解

一、极大似然估计基本思想

似然函数

- 令 $X_1,...,X_N$ 为IID，其pdf为 $p(x|\theta)$，似然函数定义为

$$\mathcal{L}(\theta)=p(\mathcal{D}|\theta)=\prod_{i=1}^{N}p(x_i|\theta)$$

  - 有时也记为 $\mathcal{L}(\theta|\mathcal{D})$，表示似然函数为在给定数据 $\mathcal{D}$的情况下，参数θ的函数。

- 似然函数在数值上是数据的联合密度，但它是参数θ的函数，$\mathcal{L}:\Theta\rightarrow[0,\infty)$。因此似然函数通常不满足密度函数的性质，如它对$\theta$的积分不必为1。

极大似然估计

- 极大似然估计（MLE）$\hat{\theta}$ 是使得 $\mathcal{L}(\theta)$ 最大的 θ，即

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta)$$

- log似然函数定义为：$l(\theta) = \log \mathcal{L}(\theta)$ ，它和似然函数在相同的位置取极大值。
  - 在不引起混淆的情况下，有时记log似然函数为似然函数
  - 相差常数倍也不影响似然函数取极大值的位置，因此似然函数中的常数项也可以抛弃。
- 在分类中 log似然有时亦称为交叉熵(cross-entropy)

负log似然可作为损失函数

- 我们可将极大似然估计套入最小化损失框架
- 因为极大

$$l(\theta)=\sum_{i=1}^{N}\log p(x_i\,|\,\theta)$$

- 等价于最小

$$-l(\theta)=\sum_{i=1}^{N}\underbrace{-\log p(x_i\,|\,\theta)}_{\text{损失函数}}$$

- 即损失函数为负log似然，然后训练集上平均损失最小

二、常见分布的参数估计

高斯分布

$$\mathcal{N}(x\,|\,\mu,\sigma^2)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

令 $x_1,\ldots,x_N\sim\mathcal{N}\left(\mu,\sigma^2\right)$，参数为 μ,σ^2，似然函数为

$$\begin{aligned}l(\mu,\sigma)&=\sum_{i=1}^{N}\log p(x_i\,|\,\mu,\sigma)\\&=-\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2-\frac{N}{2}\log\sigma^2-\frac{N}{2}\log(2\pi)\\&=-\frac{NS^2}{2\sigma^2}-\frac{N(\bar{x}-\mu)^2}{2\sigma^2}-N\log\sigma-\frac{N}{2}\log(2\pi)\end{aligned}$$

- 其中 $\bar{x}=\frac{1}{N}\sum_{i=1}^{N}x_i$ 为样本均值，$S^2=\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2$ 为样本方差

因为 $\sum_{i=1}^{N}(x_i-\mu)^2=\sum_{i=1}^{N}(x_i-\bar{x}+\bar{x}-\mu)^2=NS^2+N(\bar{x}-\mu)^2$

► 高斯分布（ cont . ）

log似然函数为 $l(\mu,\sigma)=-\frac{NS^2}{2\sigma^2}-\frac{N(\bar{x}-\mu)^2}{2\sigma^2}-N\log\sigma-\frac{N}{2}\log(2\pi)$

解方程

$$\begin{cases}\frac{\partial l(\mu,\sigma)}{\partial\mu}=\frac{N(\bar{x}-\mu)}{\sigma^2}=0\\ \frac{\partial l(\mu,\sigma)}{\partial\sigma}=-\frac{N}{\sigma}+\frac{NS^2}{\sigma^3}=0\end{cases}$$

极值点：一阶导数为0

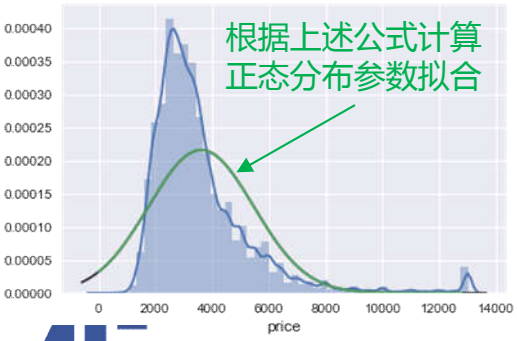
得到

$$\begin{cases}\hat{\mu}=\bar{x}=\frac{1}{N}\sum_{i=1}^{N}x_i\\ \hat{\sigma}^2=S^2=\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2=\left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right)-\bar{x}^2\end{cases}$$

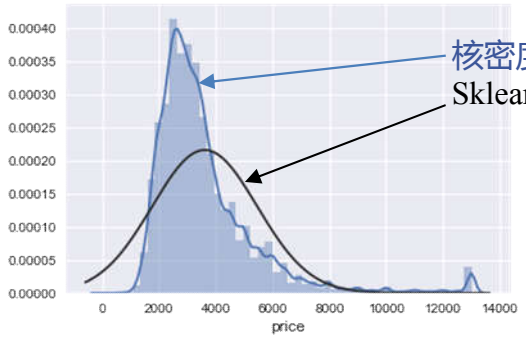
可以证明，这是似然函数的全局最大值。

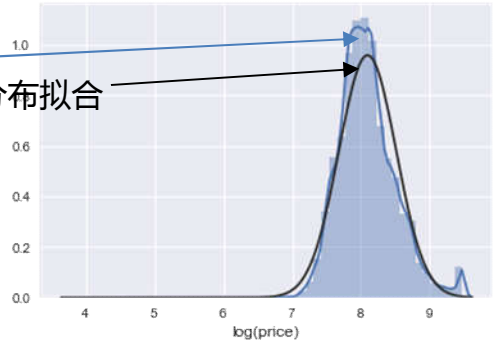
例：Rent Listing Inquires数据的price特征

- 从price特征的直方图来看，这是一个右斜的分布，可能用正态分布拟合不是不太好，不过也可以试试
  - 正态分布的均值/众数和直方图匹配不好（一些较大的样本值将均值拉大）
- 从log(price)的直方图来看，和正态分布拟合可能会好
- 注意观察非参数估计与参数估计的区别（描述参数分布需要的参数少）



sns.distplot(train.price.values, kde=True, fit=stats.norm)





sns.distplot(np.log1p(train["price"]), kde=True, fit=stats.norm)

- #核密度估计
- sns.distplot(train.price.values, kde=True, fit=stats.norm); #fit拟合
- plt.xlabel('price', fontsize=12)

- #极大似然估计，正态分布参数
- price_mean = train.price.mean()
- price_std = train.price.std()

- #显示估计的正态分布pdf
- x = np.arange(0, 5*price_std+price_mean,0.1*price_std)
- y = stats.norm.pdf(x,price_mean,price_std)
- plt.plot(x, y)
- plt.show()

Bernoulli分布

$$Ber(x|\theta)=\theta^{x}(1-\theta)^{1-x}$$

假设我们投掷硬币N次，并记录每次投掷结果的序列，用 $\mathcal{D}=\{x_1,\ldots,x_N\}$ 表示，则概率函数为 $Ber(x_i|\theta)$

似然函数为 $l(\theta)=\sum_{i=1}^{N}\log Ber(x_i|\theta)$

$$=\sum_{i=1}^{N}\log\left(\theta^{x_i}(1-\theta)^{1-x_i}\right)=N_1\log\theta+N_2\log(1-\theta)$$

其中 $\begin{cases} N_1=\sum_{i=1}^{N}x_i, & \text{试验中结果为1的次数} \\ N_2=\sum_{i=1}^{N}(1-x_i), & \text{试验中结果为0的次数} \end{cases}$

所以 $\frac{\partial l(\theta)}{\partial\theta}=\frac{N_1}{\theta}-\frac{N_2}{1-\theta}=0 \quad\Rightarrow\quad \hat{\theta}=\frac{N_1}{N_1+N_2}=\frac{N_1}{N}$

Binomial分布

$$\mathrm{Bin}(x \mid n, \theta) = \binom{n}{x} \theta^x (1-\theta)^{n-x}$$

共进行N次试验，第i次试验中抛掷了n_i次硬币

则似然函数为 $\mathcal{L}(\theta) = \prod_{i=1}^{N} Bin(x_i \mid \theta, n_i)$

$$= \prod_{i=1}^{N} \binom{n_i}{x_i} \theta^{x_i} (1-\theta)^{n_i - x_i} \propto \theta^{N_1} (1-\theta)^{N_2}$$

其中 $\begin{cases} N_1 = \sum_{i=1}^{N} x_i, \\ N_2 = \sum_{i=1}^{N} (n_i - x_i), \end{cases}$

MLE与Bernoulli分布的估计一样。

Multinoulli与Moltinomial

$$\mathrm{Mu}(x|n,\boldsymbol{\theta})=\binom{n}{x_1\ldots x_K}\prod_{k=1}^{K}\theta_k^{x_k},\quad \binom{n}{x_1\ldots x_K}\triangleq\frac{n!}{x_1!\ldots x_K!}$$

- 假设我们投掷一个有K面的骰子，共进行了N次试验，并记每次投掷结果的序列，用 $\mathcal{D}=\{x_1,\ldots,x_N\}$ 表示，$x_i\in\{1,..,K\}$
- 则似然函数为: $l(\theta)=\log p(\mathcal{D}|\theta)=\sum_{k=1}^{K}N_k\log\theta_k$
- 其中 $N_k=\sum_{i=1}^{N}\mathbb{I}(x_i=k)$ 表示N此试验中出现k的次数
- 这是带有约束 $\sum_{k=1}^{K}\theta_k=1$ 的优化问题
- 采用拉格朗日乘子法，得到

$$l(\theta,\lambda)=\sum_{k=1}^{K}N_k\log\theta_k+\lambda\left(1-\sum_{k=1}^{K}\theta_k\right)$$

Multinoulli与Moltinomial (cont.)

- 目标函数为：$l(\theta,\lambda)=\sum_{k=1}^{K}N_k\log\theta_k+\lambda\left(1-\sum_{k=1}^{K}\theta_k\right)$

- 分别对λ和θ_k求偏导并令其等于0，得到

$$\begin{cases}\dfrac{\partial l(\theta,\lambda)}{\partial\lambda}=1-\sum_{k=1}^{K}\theta_k=0\\ \dfrac{\partial l(\theta_k,\lambda)}{\partial\theta}=\dfrac{N_k}{\theta_k}-\lambda=0 \quad\Rightarrow\quad N_k=\lambda\theta_k\end{cases}\Rightarrow\boxed{\hat{\theta}_k=\frac{N_k}{N}}$$

Bag of Words语言模型

Bag of Words模型在计算机视觉中也有重要应用

- Multinomial可用于语言建模：文档由词语构成
- 词袋(Bag of Words, BoWs)模型：假设第i个词 $x_i \in \{1,...,K\}$ 是从分布 $\text{Cat}(\boldsymbol{\theta})$ 独立采样（词语相互独立，在文档分类中很合理的假设）
- 例：假设词典为

```
mary lamb little big fleece white black snow rain unk
1    2    3      4   5      6     7     8    9    10
```

- 给定序列

```
Mary had a little lamb, little lamb, little lamb,
Mary had a little lamb, its fleece as white as snow
```

- 得到每个单词的词频(直方图)为$\text{Cat}(\boldsymbol{\theta})$中$\boldsymbol{\theta}$的估计

Token	1	2	3	4	5	6	7	8	9	10
Word	mary	lamb	little	big	fleece	white	black	snow	rain	unk
Count	2	4	4	0	1	1	0	1	0	4

Bag of Words语言模型 (cont.)

- sklearn.feature_extraction.text：文本特征向量转化模块，Bag of Words模型，计算语料库中每个词的词频
  - from sklearn.feature_extraction.text import CountVectorizer
  - vec = CountVectorizer()
  - X_train = vec.fit_transform(X_train)
  - X_test = vec.transform(X_test)

特征编码流程：

1. 初始化 / 构造编码器
2. 用训练集训练（fit），并对训练集编码（transform）
3. 对测试集编码（transform）

三、机器学习模型的参数估计

回归

- 正态分布可用于回归系统噪声建模
- 回归是监督学习问题，输入$\mathcal{D}=\{(\mathbf{x}_i,y_i)\}_{i=1}^N$，输出$y$为连续型变量，学习映射 $f:\mathcal{X}\to\mathcal{Y}$
  - $y=f(\mathbf{x})+\varepsilon$ ，假设残差 $\varepsilon\sim\mathcal{N}(0,\sigma^2)$
  - 因此有$y|\mathbf{x}\sim\mathcal{N}(y|f(\mathbf{x}),\sigma^2)$

- 测试：对一个新的样本$\mathbf{x}$，预测其输出$\hat{y}=f(\mathbf{x})$，即正态分布$y|\mathbf{x}\sim\mathcal{N}(y|f(\mathbf{x}),\sigma^2)$的期望。

线性回归

- 最简单的回归模型是线性模型，即

$$y = f(\mathbf{x}) + \varepsilon$$

$$= \mathbf{w}^T \mathbf{x} + \varepsilon \qquad \text{输入的线性函数}$$

截距项

$$= w_0 + \sum_{j=1}^{D} w_j x_j + \varepsilon$$

- 其中**w**称为权重向量，ε为线性预测和真值之间的残差
- 由于$y \mid \mathbf{x} \sim \mathcal{N}(f(\mathbf{x}), \sigma^2)$，则 $p(y \mid \mathbf{x}, \boldsymbol{\theta}) \sim \mathcal{N}(y \mid \mathbf{w}^T \mathbf{x}, \sigma^2)$
- 其中模型的参数为 $\boldsymbol{\theta} = (\mathbf{w}, \sigma^2)$

线性回归的MLE

- 极大似然估计定义为 $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta})$
- 其中似然函数为

$$l(\boldsymbol{\theta}) = \log p(\mathcal{D} \mid \boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(y_i \mid x_i, \boldsymbol{\theta})$$

- 极大似然可等价地写成极小负log似然损失(negative log likelihood，NLL)

$$NLL(\boldsymbol{\theta}) = \sum_{i=1}^{N} \underbrace{-\log p(y_i \mid x_i, \boldsymbol{\theta})}_{\text{损失函数}}$$

  - 数学上的优化问题为求函数的极小值
  - MLE等价于最小经验风险

线性回归的MLE (cont.)

- 将概率模型 $p\left(y_i \mid \mathbf{x}_i, \mathbf{w}, \sigma^2\right) = \mathcal{N}\left(y_i \mid \mathbf{w}^T \mathbf{x}_i, \sigma^2\right)$ 代入，
- 似然函数为

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log\left[\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}\left(\left(y_i - \mathbf{w}^T \mathbf{x}_i\right)^2\right)\right)\right]$$

$$= -\frac{N}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\underbrace{\sum_{i=1}^{N}\left(y_i - \mathbf{w}^T \mathbf{x}_i\right)^2}_{RSS(\mathbf{w})}$$

- 其中*RSS*表示残差平方和(residual sum of squares)，*RSS*/*N* 为平均平方误差(MSE)，也可以写成残差向量的L2模，即

$$RSS(\mathbf{w}) = \|\boldsymbol{\varepsilon}\|_2^2 = \sum_{i=1}^{N} \varepsilon_i^2, \quad \varepsilon_i = \left(y_i - \mathbf{w}^T \mathbf{x}_i\right)$$

  - 损失函数为L2损失

MLE的推导

$$l(\boldsymbol{\theta}) = -\frac{N}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2$$

- 将NLL写成矩阵形式

$$NLL(\mathbf{w},\sigma) = \frac{N}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2$$

$$= \frac{N}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}\left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)^T\left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)$$

- 只取与w有关的项，得到

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

$$\frac{\partial}{\partial\mathbf{y}}\left(\mathbf{y}^T\mathbf{A}\mathbf{y}\right) = \left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{y}$$

$$\frac{\partial}{\partial\mathbf{a}}\left(\mathbf{b}^T\mathbf{a}\right) = \mathbf{b}$$

- 求导

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

Normal Equations

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

(ordinary least squares, OLS)

MLE的推导(cond.)

$$l(\boldsymbol{\theta}) = -\frac{N}{2}\log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(y_i - \mathbf{w}^T\mathbf{x}_i\right)^2$$

- 对参数σ，

$$NLL\left(\hat{\mathbf{w}},\sigma\right) = \frac{N}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\hat{\mathbf{w}}\right)^T\left(\mathbf{y}-\mathbf{X}\hat{\mathbf{w}}\right)$$

- 求导

$$\frac{\partial}{\partial\sigma}NLL(\hat{\mathbf{w}},\sigma^2) = \frac{N}{\sigma} - \frac{1}{\sigma^3}\left(\mathbf{y}-\mathbf{X}\hat{\mathbf{w}}\right)^T\left(\mathbf{y}-\mathbf{X}\hat{\mathbf{w}}\right) = 0$$

- 得到

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

当样本数目N较小时，可采用OLS结论，用矩阵QR分解分解得到优化解
当样本数目N较大时，可采用随机梯度下降方法优化求解

梯度下降

- $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} J(\mathbf{w}) = \arg\min_{\mathbf{w}} \left[\sum_{i=1}^{N} \left(f(\mathbf{x}_i) - y_i \right)^2 \right]$
- 梯度下降
  - 给定初值$\mathbf{w}^0$
  - 更新$\mathbf{w}$，使得 $J(\mathbf{w})$越来越小

$$w_d^t = w_d^{t-1} - \alpha \frac{\partial}{\partial w_d} J(\mathbf{w})$$

$$= w_d^{t-1} - \alpha \sum_{i=1}^{N} \left[2 \left(f(\mathbf{x}_i) - y_i \right) x_{id} \right]$$

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

  - $\mathbf{w}$的各维同时更新：$f(\mathbf{x}_i) = \left[\mathbf{w}^{t-1} \right]^T \mathbf{x}_i$
  - α称为学习率(Learning Rate)
    - 学习率的选择参看Standford CS229课程线性回归部分课件
  - 直到收敛到某个$\mathbf{w}$值，使得 $J(\mathbf{w})$ 最小

案例：波士顿房价分析

Regression_bostonhouseprice.ipynb

- 波士顿房屋这些数据于1978年开始统计，共506个数据点，涵盖了麻省波士顿不同郊区房屋14种特征的信息

7. Attribute Information:	
1. CRIM per capita crime rate by town	1. 城镇人均犯罪率
2. ZN proportion of residential land zoned for lots over 25,000 sq.ft.	2. 住宅用地所占比例，25000英尺
3. INDUS proportion of non-retail business acres per town	3. 城镇中非商业用地的所占比例
4. CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)	4. CHAS查尔斯河虚拟变量，用于回归分析
5. NOX nitric oxides concentration (parts per 10 million)	5. 环保指标
6. RM average number of rooms per dwelling	6. 每栋住宅的房间数
7. AGE proportion of owner-occupied units built prior to 1940	7. 1940年以前建成的自住单位的比例
8. DIS weighted distances to five Boston employment centres	8. 距离五个波士顿就业中心的加权距离
9. RAD index of accessibility to radial highways	9. 距离高速公路的便利指数
10. TAX full-value property-tax rate per $10,000	10. 每一万美元的不动产税率
11. PTRATIO pupil-teacher ratio by town	11. 城镇中教师学生比例
12. B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town	12. 城镇中黑人比例
13. LSTAT % lower status of the population	13. 地区有多少百分比的房东属于是低收入阶层
14. MEDV Median value of owner-occupied homes in $1000's	14. 自住房屋房价的中位数

案例：波士顿房价分析

- 最小二乘法回归系数

7. Attribute Information:	
1. CRIM per capita crime rate by town	1. 城镇人均犯罪率
2. ZN proportion of residential land zoned for lots over 25,000 sq.ft.	2，住宅用地所占比例，25000英尺
3. INDUS proportion of non-retail business acres per town	3，城镇中非商业用地的所占比例
4. CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)	4，CHAS查尔斯河虚拟变量，用于回归分析
5. NOX nitric oxides concentration (parts per 10 million)	5，环保指标
6. RM average number of rooms per dwelling	6，每栋住宅的房间数
7. AGE proportion of owner-occupied units built prior to 1940	7，1940年以前建成的自住单位的比例
8. DIS weighted distances to five Boston employment centres	8，距离五个波士顿就业中心的加权距离
9. RAD index of accessibility to radial highways	9，距离高速公路的便利指数
10. TAX full-value property-tax rate per $10,000	10，每一万美元的不动产税率
11. PTRATIO pupil-teacher ratio by town	11，城镇中教师学生比例
12. B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town	12，城镇中黑人比例
13. LSTAT % lower status of the population	13，地区有多少百分比的房东属于是低收入阶层
14. MEDV Median value of owner-occupied homes in $1000's	14，自住房屋房价的中位数

-0.11286566,
0.1306885 ,
0.01207992,
0.09054443,
-0.17880511,
0.31821979,
-0.01744478,
-0.33320158,
0.26716638,
-0.21737875,
-0.20384674,
0.05662515,
-0.40794066]

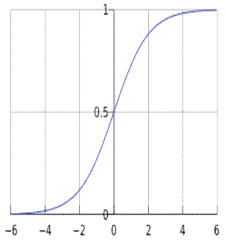
Logistic回归

- Logistic回归是线性回归的扩展，用于分类任务：
  - 1. 目标y为二值变量：因此高斯分布$p(y|\mathbf{x},\boldsymbol{\theta})$变成Bernoulli分布

$$y \in \{0,1\}, p(y \mid \mathbf{x}, \mathbf{w}) = \mathrm{Ber}\left(y|\mu(\mathbf{x})\right)$$

  - 其中 $\mu(\mathbf{x}) = \mathbb{E}(y \mid \mathbf{x}) = p(y=1 \mid \mathbf{x})$
  - 2. 计算输入$\mathbf{x}$的线性组合，但经过函数$\mu(\mathbf{x}) = sigm(\mathbf{w}^{\mathrm{T}}\mathbf{x})$ 使得$0 \le \mu(\mathbf{x}) \le 1$
  - 其中sigmoid函数（S形函数）定义为

$$sigm(\eta) = \frac{1}{1+\exp(-\eta)} = \frac{\exp(\eta)}{\exp(\eta)+1}$$



  - 亦被称为logistic函数或logit函数
  - 综合起来：$p(y \mid \mathbf{x}, \mathbf{w}) = \mathrm{Ber}\left(y \mid sigm(\mathbf{w}^{\mathrm{T}}\mathbf{x})\right)$
    - 因为和线性回归相似，因此被称为Logistic回归（虽然是分类）

为什么用logistic函数？

- 来自神经科学：
  - 神经元对其输入加权和：$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$
  - 如果该和大于某阈值，神经元发放脉冲：$f(\mathbf{x}) > \tau$
- Logstic回归：当 $p(y=1|\mathbf{x},\mathbf{w}) > p(y=0|\mathbf{x},\mathbf{w})$ 时发放
- 定义Log Odds Ratio：

$$\mathrm{LOR}(\mathbf{x}) = \log\frac{p(y=1|\mathbf{x},\mathbf{w})}{p(y=0|\mathbf{x},\mathbf{w})}$$

$$= \log\left[\frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}\frac{1+\exp(-\mathbf{w}^T\mathbf{x})}{\exp(-\mathbf{w}^T\mathbf{x})}\right]$$

$$= \log\left[\exp(\mathbf{w}^T\mathbf{x})\right] = \mathbf{w}^T\mathbf{x}$$

- 因此 *iff* $\mathrm{LOR}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} > 0$ 发放

MLE

$$\mathrm{Ber}(\theta) = \theta^{x}(1-\theta)^{1-x}$$

- Logistic回归： $p(y \mid \mathbf{x}, \mathbf{w}) = \mathrm{Ber}(y \mid \mu(x)), \quad \mu(x) = sigm(\mathbf{w}^{\mathrm{T}}\mathbf{x})$
- 负log似然为

$$J(\mathbf{w}) = NLL(\mathbf{w}) = -\sum_{i=1}^{N} \log\left[(\mu_i)^{y_i} \times (1-\mu_i)^{(1-y_i)}\right]$$

$$= -\sum_{i=1}^{N}\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i)\right]$$

  - 其中 $\mu_i = \mu(\mathbf{x}_i)$

优化求解：梯度下降 / 牛顿法

梯度下降

$$J(\mathbf{w}) = -\sum_{i=1}^{N}\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i) \right]$$

要： $\min J(\mathbf{w})$

Repeat $\{$

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w})$$

(同时更新所有 w_j)

$\}$

梯度下降

$$J(\mathbf{w}) = -\sum_{i=1}^{N}\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i) \right]$$

要 $\min J(\mathbf{w})$ ：

Repeat $\{$

$$w_j := w_j - \alpha \sum_{i=1}^{N} \left(\mu(\mathbf{x}_i) - y_i \right) x_{ij}$$

预测残差

$$\mu(\mathbf{x}) = \frac{\exp(\mathbf{w}^T\mathbf{x})}{\exp(\mathbf{w}^T\mathbf{x})+1}$$

$\}$ (同时更新所有 w_j)

算法同线性回归 $w_j := w_j - \alpha \sum_{i=1}^{N} \left(f(\mathbf{x}_i) - y_i \right) x_{ij}$ 看起来一样！

当然$f(\mathbf{x})$不同（线性回归中 $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ ）

事实上所有的线性模型的梯度下降递推公式都是如此

$$J(\mathbf{w}) = -\sum_{i=1}^{N}\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i)\right]$$

$$g(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = \sum_{i=1}^{N}\left[-y_i \times \frac{1}{\mu(\mathbf{x}_i)} \frac{\partial}{\partial \mathbf{w}} \mu(\mathbf{x}_i) + (1-y_i) \times \frac{1}{1-\mu(\mathbf{x}_i)} \frac{\partial}{\partial \mathbf{w}} \mu(\mathbf{x}_i) \right]$$

$$= \sum_{i=1}^{N}\left[-y_i \times \frac{1}{\mu(\mathbf{x}_i)} + (1-y_i) \times \frac{1}{1-\mu(\mathbf{x}_i)} \right] \frac{\partial}{\partial \mathbf{w}} \mu(\mathbf{x}_i)$$

$$= \sum_{i=1}^{N}\left[-y_i \times \frac{1}{\mu(\mathbf{x}_i)} + (1-y_i) \times \frac{1}{1-\mu(\mathbf{x}_i)} \right] \mu(\mathbf{x}_i)\left(1-\mu(\mathbf{x}_i)\right)\mathbf{x}_i$$

$$= \sum_{i=1}^{N}\left[-y_i \times \left[1-\mu(\mathbf{x}_i)\right] + (1-y_i)\,\mu(\mathbf{x}_i) \right]\mathbf{x}_i$$

$$= \sum_{i=1}^{N}\left[-y_i + \mu(\mathbf{x}_i) \right]\mathbf{x}_i$$

$$= \sum_{i=1}^{N}\left[\mu(\mathbf{x}_i) - y_i \right]\mathbf{x}_i$$

$$\frac{\partial}{\partial \mathbf{w}} \mu(\mathbf{x}) = \mu(\mathbf{x})\left(1-\mu(\mathbf{x})\right)\mathbf{x}$$

$$\mu(\mathbf{x}) = \frac{\exp\left(\mathbf{w}^T\mathbf{x}\right)}{\exp\left(\mathbf{w}^T\mathbf{x}\right)+1}$$

$$1-\mu(\mathbf{x}) = \frac{1}{\exp\left(\mathbf{w}^T\mathbf{x}\right)+1}$$

$$\frac{\partial}{\partial \mathbf{w}}\mu(\mathbf{x}) = \frac{\frac{\partial}{\partial \mathbf{w}}\left[\exp\left(\mathbf{w}^T\mathbf{x}\right)\right]\left(\exp\left(\mathbf{w}^T\mathbf{x}\right)+1\right) - \exp\left(\mathbf{w}^T\mathbf{x}\right)\frac{\partial}{\partial \mathbf{w}}\left[\exp\left(\mathbf{w}^T\mathbf{x}\right)+1\right]}{\left[\exp\left(\mathbf{w}^T\mathbf{x}\right)+1\right]^2}$$

$$= \frac{\exp\left(\mathbf{w}^T\mathbf{x}\right)\left(\exp\left(\mathbf{w}^T\mathbf{x}\right)+1\right)\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{w}^T\mathbf{x}\right) - \exp\left(\mathbf{w}^T\mathbf{x}\right)\exp\left(\mathbf{w}^T\mathbf{x}\right)\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{w}^T\mathbf{x}\right)}{\left[\exp\left(\mathbf{w}^T\mathbf{x}\right)+1\right]^2}$$

$$= \frac{\exp\left(\mathbf{w}^T\mathbf{x}\right)}{\left[\exp\left(\mathbf{w}^T\mathbf{x}\right)+1\right]^2}\mathbf{x} = \mu(\mathbf{x})\left(1-\mu(\mathbf{x})\right)\mathbf{x}$$

$$\frac{\partial}{\partial \mathbf{w}}\left(\mathbf{w}^T\mathbf{x}\right) = \frac{\partial}{\partial \mathbf{w}}\left(\mathbf{x}^T\mathbf{w}\right) = \mathbf{x}$$

$$\mu(\mathbf{x}) = \frac{\exp\left(\mathbf{w}^T\mathbf{x}\right)}{\exp\left(\mathbf{w}^T\mathbf{x}\right)+1}$$

$$1-\mu(\mathbf{x}) = \frac{1}{\exp\left(\mathbf{w}^T\mathbf{x}\right)+1}$$

（一阶）梯度下降法

- 两类logistic回归：
  - 损失函数：(负log似然)

$$J(\mathbf{w}) = -\sum_{i=1}^{N}\left[y_i \log(\mu_i) + (1-y_i)\log(1-\mu_i)\right]$$

$$\mathbf{g}(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) = \sum_{i=1}^{N}(\mu_i - y_i)\mathbf{x}_i = \mathbf{X}^T(\boldsymbol{\mu} - \mathbf{y})$$

$$\mathbf{w}^{k+1} := \mathbf{w}^k - \alpha \mathbf{g}(\mathbf{w}^k)$$

$$\frac{\partial}{\partial \mathbf{w}}\mu(\mathbf{x}) = \mu(\mathbf{x})(1-\mu(\mathbf{x}))\mathbf{x}$$

$$\mathbf{H}(\mathbf{w}) = \frac{\partial^2}{\partial \mathbf{w}^2}\left[J(\mathbf{w})\right] = \frac{\partial}{\partial \mathbf{w}}\left[\mathbf{g}(\mathbf{w})^T\right] = \sum_{i=1}^{N}\left(\frac{\partial}{\partial \mathbf{w}}\mu_i\right)\mathbf{x}_i^T$$

$$= \mu_i(1-\mu_i)\mathbf{x}_i\mathbf{x}_i^T = \mathbf{X}^T diag\left(\mu_i(1-\mu_i)\right)\mathbf{X}$$

正定矩阵，凸优化

牛顿法

- 亦称牛顿-拉夫逊（ Newton-Raphson ）方法
  - 牛顿在17世纪提出的一种近似求解方程的方法
  - 使用函数$f(x)$的泰勒级数的前面几项来寻找方程 $f(x)=0$ 的根

- 在求极值问题中，求 $\mathbf{g}(\mathbf{w})=\frac{\partial}{\partial \mathbf{w}}J(\mathbf{w})=0$ 的根
  - 对应处$J(\mathbf{w})$ 取极值

牛顿法

- 将导数$\mathbf{g}(\mathbf{w})$ 在 $\mathbf{w}^t$ 处进行Taylor展开：

$$0 = \mathbf{g}(\hat{\mathbf{w}}) = g(\mathbf{w}^t) + (\hat{\mathbf{w}} - \mathbf{w}^t)\mathbf{H}(\mathbf{w}^t) + Op(\hat{\mathbf{w}} - \mathbf{w}^t)$$

- 从而得到

$$\hat{\mathbf{w}} \approx \mathbf{w}^t - \mathbf{H}^{-1}(\mathbf{w}^t)g(\mathbf{w}^t)$$

- 因此迭代机制为：

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \mathbf{H}^{-1}(\mathbf{w}^t)g(\mathbf{w}^t)$$

  - 也被称为二阶梯度下降法，移动方向：$\mathbf{H}(\mathbf{w}^t)\mathbf{d} = -\mathbf{g}(\mathbf{w}^t)$
  - Vs. 一阶梯度法，移动方向：$\mathbf{d} = -\mathbf{g}(\mathbf{w}^t)$　　移动

Iteratively Reweighted Least Squares

$$\mathbf{g}^t(\mathbf{w}) = \mathbf{X}^T\left(\boldsymbol{\mu}^t - \mathbf{y}\right)$$

$$\mathbf{H}^t(\mathbf{w}) = \mathbf{X}^T\mathbf{S}^t\mathbf{X}$$

$$\mathbf{S}^t := \mathrm{diag}\left(\mu_1^t\left(1-\mu_1^t\right),\ldots,\mu_N^t\left(1-\mu_N^t\right)\right)$$

$$\mu_i^t = \mathrm{sigm}\left(\left(\mathbf{w}^t\right)^T\mathbf{x}_i\right)$$

$$\begin{aligned}\mathbf{w}^{t+1} &= \mathbf{w}^t - \left(\mathbf{H}^t\right)^{-1}\mathbf{g}^t \\ &= \mathbf{w}^t + \left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)^{-1}\mathbf{X}^T\left(\mathbf{y}-\boldsymbol{\mu}^t\right) \\ &= \left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)^{-1}\left[\left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)\mathbf{w}^t + \mathbf{X}^T\left(\mathbf{y}-\boldsymbol{\mu}^t\right)\right] \\ &= \left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)^{-1}\mathbf{X}^T\left[\mathbf{S}^t\mathbf{X}\mathbf{w}^t + \mathbf{y} - \boldsymbol{\mu}^t\right]\end{aligned}$$

Rewrite as a weighted least squares problem: 最小化

$$\sum_{i=1}^{N} S_i^t\left(z_i^k - \mathbf{w}^T\mathbf{x}_i\right)$$

其中

$$\hat{\mathbf{w}} = \left(\mathbf{X}^T\mathbf{S}^t\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{S}^t\mathbf{z}^t$$

$$\mathbf{z}^t = \mathbf{X}\mathbf{w}^t + \left(\mathbf{S}^t\right)^{-1}\left(\mathbf{y}-\boldsymbol{\mu}^t\right)$$

$\mathbf{S}^t$ is diagonal → $\mathbf{z}^t$ can be rewrite in component form

$$\mathbf{z}_i^t = \left(\mathbf{w}^t\right)^T\mathbf{x}_i + \frac{y_i - \mu_i^t}{\mu_i^t\left(1-\mu_i^t\right)}$$

Iteratively Reweighted Least Squares (cond)

- Iteratively reweighted least squares(IRLS)

1 $\mathbf{w} = \mathbf{0}_D$

2 $w_0 = \log(\bar{y} / (1 - \bar{y}))$

3 **repeat**

4 $\eta_i = w_0 + \mathbf{w}^T \mathbf{x}_i$

5 $\mu_i = \text{sigm}(\eta_i)$

6 $s_i = \mu_i (1 - \mu_i)$

7 $z_i = \eta_i + \frac{y_i - \mu_i}{s_i}$

8 $\mathbf{S} = \text{diag}(s_{1:N})$

9 $\mathbf{w} = \left(\mathbf{X}^T \mathbf{S} \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{S} \mathbf{z}$

10 **until** *converged*

$$\mathbf{S} = \text{diag}\left(\mu_1 \left(1 - \mu_1\right), \ldots, \mu_N \left(1 - \mu_N\right)\right)$$

$$\mathbf{z}_i = \mathbf{w}^T \mathbf{x}_i + \frac{y_i - \mu_i}{\mu_i \left(1 - \mu_i\right)}$$

Weighted least square

► 例：Titanic存活预测

- Titanic.ipynb

► 朴素贝叶斯 (Naive Bayes Classifier, NBC)

- 假设共有 C 个类别 $y \in (1, 2, \ldots, C)$
- 每个类别有特征 $\mathbf{x} = (x_1, x_2, \ldots, x_D)$
- 朴素贝叶斯分类器是比较简单也很常用的分类器
  - 简单/朴素：假设各维特征在给定类别标签的情况下条件独立

$$p(\mathbf{x} \mid y = c, \theta) = \prod_{j=1}^{D} p(x_j \mid y = c, \theta)$$

  - 通常即使特征条件独立的假设不满足，NBC在实际系统 中的性能也不错。因为NBC比较简单，不容易过拟合（如特征为Bernoulli分布的话，只需$O(CD)$个参数）

其中$\boldsymbol{\pi}$、$\boldsymbol{\theta}$分别为y的先验分布和类条件分布的参数

- 单个数据点的概率为

$$p(\mathbf{x}_i, y_i \mid \boldsymbol{\theta}, \boldsymbol{\pi}) = p(\mathbf{x}_i \mid y_i, \boldsymbol{\theta})\, p(y_i \mid \boldsymbol{\pi}) \quad \text{链规则}$$

$$= p(y_i \mid \boldsymbol{\pi}) \prod_{j=1}^{D} p(x_{ij} \mid y_i, \boldsymbol{\theta}_j) \quad \text{条件独立}$$

$$= \prod_{c} \pi_c^{\mathbb{I}(y_i=c)} \prod_{j=1}^{D} \prod_{c} p(x_{ij} \mid \theta_{jc})^{\mathbb{I}(y_i=c)} \quad y_i \sim Cat(y \mid \boldsymbol{\pi})$$

- 所以log似然为

$$l(\boldsymbol{\theta}) = \log p(\mathcal{D} \mid \boldsymbol{\theta}) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log p(x_{ij} \mid \theta_{jc})$$

N_c：$y=c$的样本数

$i:y_i=c$：$y=c$的样本

NBC

- 目标函数为

$$l(\theta)=\log p(\mathcal{D}|\theta)=\sum_{c=1}^{C}N_c\log\pi_c+\sum_{j=1}^{D}\sum_{c=1}^{C}\sum_{i:y_i=c}\log p(x_{ij}|\theta_{jc})$$

- 可分别优化$\boldsymbol{\pi}$和$\boldsymbol{\theta}$
- 根据根据之前对multinomial/Cat分布的讨论，

$$y_i\sim Cat(y|\boldsymbol{\pi})\Rightarrow \boxed{\hat{\pi}_c=\frac{N_c}{N}}$$

- 其中 $N_c=\sum_i \mathbb{I}(y_i=c)$

NBC – 二值特征

$$\hat{\pi}_c = \frac{N_c}{N}$$

- 目标函数为

$$l(\theta) = \log p(\mathcal{D} \mid \theta) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log p(x_{ij} \mid \theta_{jc})$$

- 若 $p(\mathbf{x}_j \mid y = c) \sim Ber(\theta_{jc})$, 可得到

$$\hat{\theta}_{jc} = \frac{N_{jc}}{N_c}$$

- 其中 $N_c = \sum_i \mathbb{I}(y_i = c), \quad N_{jc} = \sum_i \mathbb{I}(x_{ij} = 1, y_i = c)$

NBC MLE – 多值离散特征

$$\hat{\pi}_c = \frac{N_c}{N}$$

- 目标函数为

$$l(\theta) = \log p(\mathcal{D}|\theta) = \sum_{c=1}^{C} N_c \log \pi_c + \sum_{j=1}^{D} \sum_{c=1}^{C} \sum_{i:y_i=c} \log p(x_{ij}|\theta_{jc})$$

- 若 $p(x_j|y=c) \sim Cat(\boldsymbol{\theta}_{jc})$ ，可得到

$$\hat{\theta}_{jck} = \frac{N_{jck}}{\sum_{k'} N_{jck'}} = \frac{N_{jck}}{N_{jc}}$$

- 其中 $N_{jck} = \sum_i \mathbb{I}(x_{ij} = k, y_i = c)$

NBC MLE – 连续特征

- 假设 $x_j \mid y=c \sim \mathcal{N}\left(\mu_c, \sigma_c^2\right)$
- 则 $\hat{\mu}_{jc} = \dfrac{\sum_{i:y_i=c} x_{ij}}{N_c},$

$$\hat{\sigma}_{jc}^2 = \frac{\sum_{i:y_i=c}\left(x_{ij}-\hat{\mu}_{jc}\right)^2}{N_c}$$

sklearn 支持上述三种朴素贝叶斯实现：http://scikit-learn.org/stable/modules/naive_bayes.html
BernoulliNB、MultinomialNB、GaussianNB

用NBC进行预测

- 预测为：$p(y=c|\mathbf{x},\mathcal{D}) \propto p(y=c|\mathcal{D})\prod_{j=1}^{D} p(x_j|y=c,\mathcal{D})$
- 将给定数据条件$\mathcal{D}$换成参数的MLE插入，得到

$$p(y=c|\mathbf{x},\mathcal{D}) \propto p(y=c|\mathcal{D})\prod_{j=1}^{D} p(x_j|y=c,\mathcal{D})$$

假设类条件为Bernoulli分布：

$$\propto \mathrm{Cat}(y=c|\hat{\boldsymbol{\pi}})\prod_{j=1}^{D} Ber(x_j|\hat{\theta}_{jc})$$

$$=\hat{\pi}_c\prod_{j=1}^{D}\left(\hat{\theta}_{jc}\right)^{\mathbb{I}(x_j=1)}\left(1-\hat{\theta}_{jc}\right)^{\mathbb{I}(x_j=0)}$$

案例：新闻文档分类

- NBC_News.ipynb

► 例：Titanic生存预测

- Titanic.ipynb

四、估计的评价

估计量的评价标准

- 一个好的估计有什么性质?
- 无偏性
  - 估计的偏差（bias）为
    $$bias\left(\hat{\theta}\right)=\mathbb{E}_{\mathcal{D}}\left(\hat{\theta}\right)-\theta$$
  - 若 $\mathbb{E}_{\mathcal{D}}\left(\hat{\theta}\right)=\theta$ ，则该估计是无偏估计。
- 相容性
  - 若 $\hat{\theta}\xrightarrow{P}\theta$ ，则该点估计是相容(consistent)的。
- 有效性
  - 无偏估计中，方差较小的一个更有效（收敛速度更快）

对分布 $p\left(x_1,\ldots,x_N\mid\theta\right)=\prod_{i=1}^{N}p\left(x_i\mid\theta\right)$ 求期望，而不是对 θ 平均

MLE的性质

- 优点:
  - 简单，有时有解析解
  - 有一些好的理论性质 : 渐近相容性、渐近无偏性、渐近有效性

- 缺点:
  - 过拟合
  - 没有非确定表示：点估计
    - 渐近正态分布
    - 抽样分布

Overfitting

- 如在Titanic数据中，训练样本里面有特征SibSp表示姐妹兄弟数
  - SibSp=7的样本只有一个，该样本存活 → SibSp=7活着的概率等于100%
  - 若SibSp=9的样本也只有一个，并且死了→ SibSp=9的样本生存的概率是0%
  - 假如某个人的某个特征的最大似然概率是0，那么他的整个乘积也是0

- 与黑天鹅悖论(black swan paradox)类似
- 解决方案
  - 将计算概率的分子分母都适当扩大
  - 对离散型特征，区间 / 类别适当划分（合并）

偏差—方差分解

- 点估计的性能有时通过均方误差(MSE, mean squared error)来评价：

$$MSE = \mathbb{E}_{\mathcal{D}}\left(\hat{\theta} - \theta\right)^2$$

- MSE可分解为: $MSE = bias\left(\hat{\theta}\right)^2 + \mathbb{V}_{\mathcal{D}}\left(\hat{\theta}\right)$

估计的偏差/正确性　　估计的变化程度/精度

- 其中偏差为 $bias\left(\hat{\theta}\right) = \mathbb{E}_{\mathcal{D}}\left(\hat{\theta}\right) - \theta$
- 如果bias=0，我们称其为无偏估计，此时 $MSE = \mathbb{V}_{\mathcal{D}}\left(\hat{\theta}\right)$ 。
- 所以为了使估计的MSE小，估计的偏差和方差都要小

▶ 证明：

证明： $MSE = bias\left(\hat{\theta}\right)^2 + \mathbb{V}_{\mathcal{D}}\left(\hat{\theta}\right)$

令 $\bar{\theta} = \mathbb{E}_{\mathcal{D}}\left(\hat{\theta}\right)$，则

$$MSE = \mathbb{E}_{\mathcal{D}}\left(\hat{\theta} - \theta\right)^2 = \mathbb{E}_{\mathcal{D}}\left(\hat{\theta} - \bar{\theta} + \bar{\theta} - \theta\right)^2$$

$$= \mathbb{E}_{\mathcal{D}}\left(\hat{\theta} - \bar{\theta}\right)^2 + \mathbb{E}_{\mathcal{D}}\left(\bar{\theta} - \theta\right)^2 + 2\left(\bar{\theta} - \theta\right)\mathbb{E}_{\mathcal{D}}\left(\hat{\theta} - \bar{\theta}\right)$$

$$= \mathbb{E}_{\mathcal{D}}\left(\hat{\theta} - \bar{\theta}\right)^2 + \left(\bar{\theta} - \theta\right)^2 + 2\left(\bar{\theta} - \theta\right)\left(\bar{\theta} - \bar{\theta}\right)$$

$$= \mathbb{V}_{\mathcal{D}}\left(\hat{\theta}\right) + bias\left(\hat{\theta}\right)^2$$

► 例：Bernoulli分布

对Bernoulli分布，参数的MLE为 $\hat{\theta}=\frac{1}{N}\sum_{i=1}^{N}X_i$

由于 $X_i \sim Ber(\theta)$ ，$S=\sum_{i=1}^{N}X_i \sim Bin(N,\theta)$

则抽样分布为

$$p(\hat{\theta})=p(S=N\hat{\theta}) \sim Bin(N\hat{\theta}\,|\,N,\theta)$$

则该分布的期望为 $\mathbb{E}(\hat{\theta})=\frac{1}{N}\mathbb{E}(S)=\frac{1}{N}\times N\theta=\theta$

方差为 $\mathbb{V}(\hat{\theta})=\mathbb{V}\left(\frac{1}{N}\sum_{i=1}^{N}X_i\right)=\frac{1}{N^2}\mathbb{V}\left(\sum_{i=1}^{N}X_i\right)=\frac{1}{N^2}\left(\sum_{i=1}^{N}\theta(1-\theta)\right)=\frac{\theta(1-\theta)}{N}$

样本数N越大，估计方差越小

▶例：Bernoulli分布

标准误差可用插入估计近似为 $\widehat{se}=\sqrt{\frac{\hat{\theta}\left(1-\hat{\theta}\right)}{N}}$

当N足够大时，二项分布可用高斯分布近似

$$p\left(\hat{\theta}\right)\approx\mathcal{N}\left(\hat{\theta}\,|\,\theta,se^2\right)$$

亦可记为 $\left(\hat{\theta}-\theta\right)/se \quad \mathcal{N}\left(0,1\right)$

我们称之为渐近正态(Asymptotically Normal)的。 （任何分布均可）

监督学习模型的偏差-方差分解

- 以回归任务为例, 学习算法 f 的平方预测误差期望为:

$$
\begin{aligned}
E(f;D) =& \mathbb{E}_D\left[(f(\boldsymbol{x};D)-y_D)^2\right] \\
=& \mathbb{E}_D\left[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x})+\bar{f}(\boldsymbol{x})-y_D)^2\right] \\
=& \mathbb{E}_D\left[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2\right]+\mathbb{E}_D\left[(\bar{f}(\boldsymbol{x})-y_D)^2\right] \\
&+\mathbb{E}_D\left[2\left(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x})\right)\left(\bar{f}(\boldsymbol{x})-y_D\right)\right] \\
=& \mathbb{E}_D\left[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2\right]+\mathbb{E}_D\left[(\bar{f}(\boldsymbol{x})-y_D)^2\right] \\
=& \mathbb{E}_D\left[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2\right]+\mathbb{E}_D\left[(\bar{f}(\boldsymbol{x})-y+y-y_D)^2\right] \\
=& \mathbb{E}_D\left[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2\right]+\mathbb{E}_D\left[(\bar{f}(\boldsymbol{x})-y)^2\right]+\mathbb{E}_D\left[(y-y_D)^2\right] \\
&+2\mathbb{E}_D\left[(\bar{f}(\boldsymbol{x})-y)\left(y-y_D\right)\right] \\
=& \mathbb{E}_D\left[(f(\boldsymbol{x};D)-\bar{f}(\boldsymbol{x}))^2\right]+(\bar{f}(\boldsymbol{x})-y)^2+\mathbb{E}_D\left[(y-y_D)^2\right]
\end{aligned}
$$

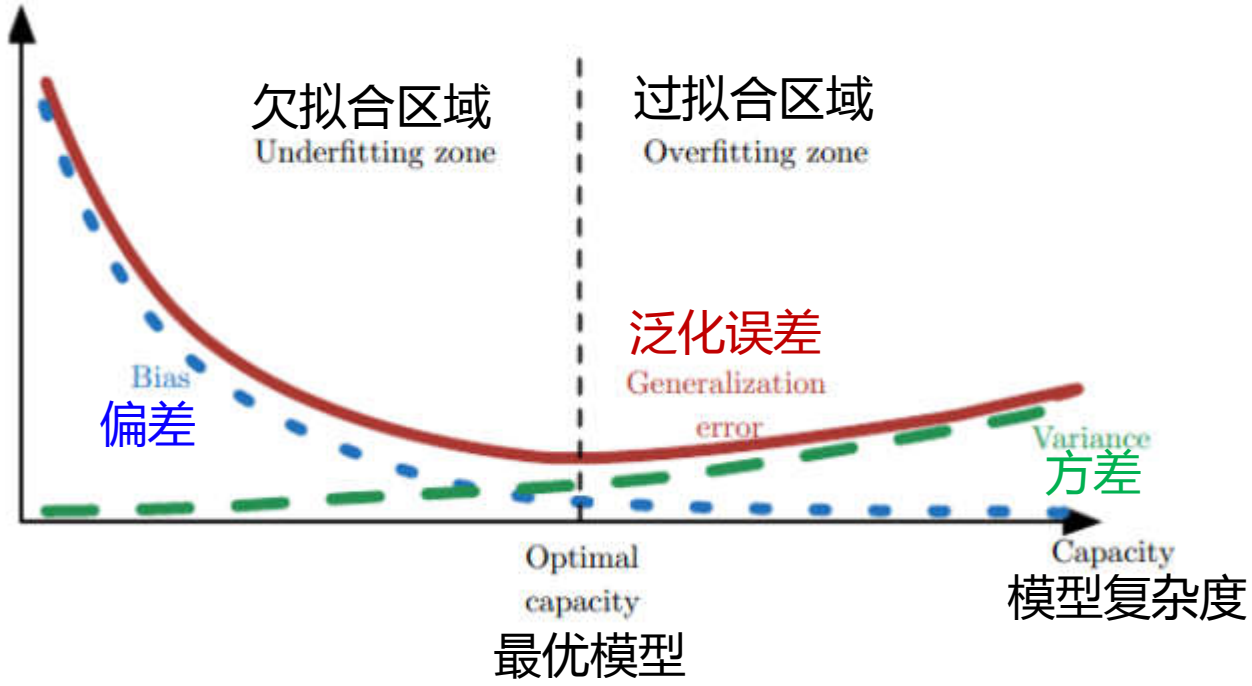
方差　　偏差　　系统噪声

•偏差：学习算法的期望预测与真实结果的偏离程序, 即 **刻画了学习算法本身的拟合能力** .

•方差：同样大小的训练集的变动所导致的学习性能的变化, 即 **刻画了数据扰动所造成的影响** .

•噪声：在当前任务上任何学习算法所能达到的期望泛化误差的下界, 即 **刻画了学习问题本身的难度** .

► 偏差-方差平衡

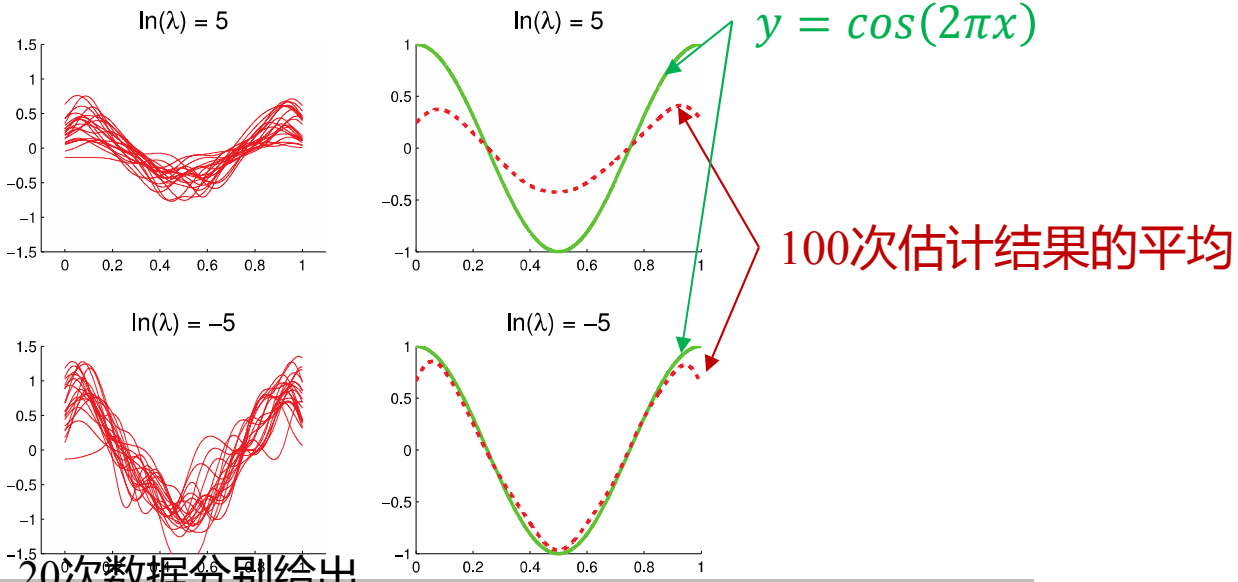

欠拟合区域
Underfitting zone
过拟合区域
Overfitting zone
泛化误差
Generalization
error
Bias
偏差
Variance
方差
Optimal
capacity
最优模型
Capacity
模型复杂度

▶例：偏差-方差平衡

- 从$y = cos(2\pi x) + \varepsilon, \varepsilon \sim N(0, 0.1^2)$每次产生$N$=25个样本，共产生$B$=100次，采用岭回归估计曲线

$ln(\lambda) = 5$: 正则大，模型简单，偏差大，方差小

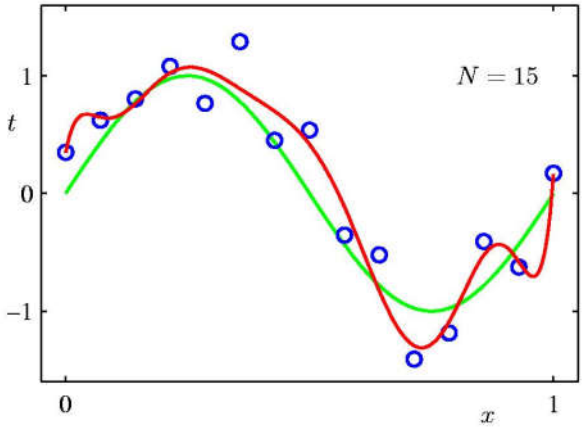
$ln(\lambda) = -5$: 正则小，模型复杂，偏差小，方差大



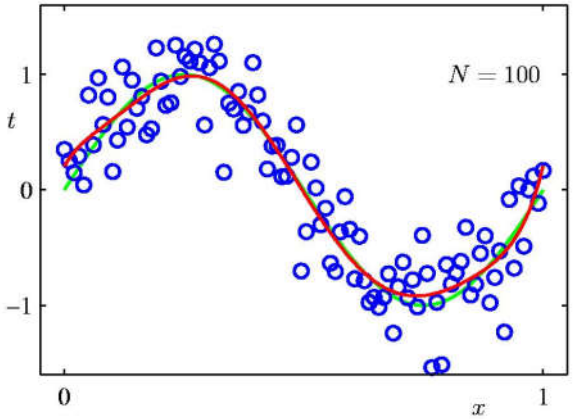
20次数据分别给出的估计（20条线）

当数据更多时

- 当数据更多时，可考虑更复杂的模型
- 例：Sin曲线拟合：



9阶多项式拟合，训练样本数N=15

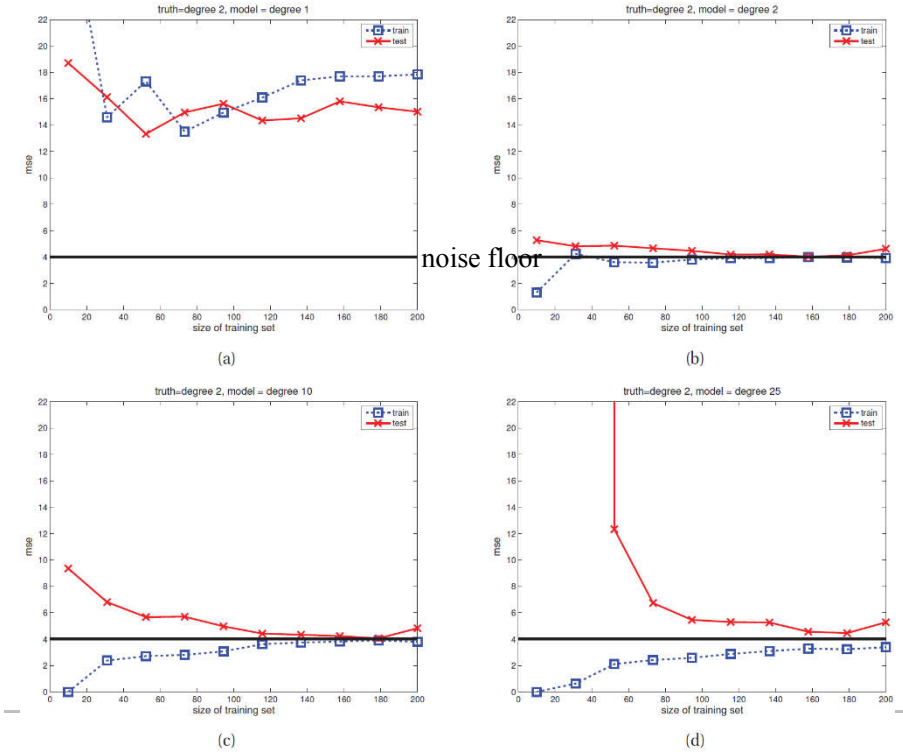


9阶多项式拟合，训练样本数N=100

65

当数据更多时(cont.)

- 例：用多项式拟合二阶多项式



noise floor

测试误差由两部分组成：

(1)各个模型都会有的不可约部分：数据产生过程的变化(noise floor)
(2)模型与数据产生过程的差异(structure error)

当模型足够复杂，可建模数据产生过程，
当$N \to \infty$，测试误差→ noise floor
但简单模型趋近的速度更快

66

MLE的渐近正态性

Sir Ronald Aylmer Fisher
(1890-1962)

- 渐近正态性：当样本数目足够多时，MLE估计的分布是正态分布

- 为了证明这一性质，引入记分函数和Fisher信息
  - 略，参见All of statistics 第9.7节

- 当记分函数和Fisher信息的形式比较简单时，可解析求解
- 若解析计算困难，可用参数bootstrap方法计算

估计的抽样分布

- 我们用 $\hat{\theta}=\hat{\theta}(\mathcal{D})$ 表示我们根据观测数据$\mathcal{D}$得到的对参数θ的估计，该参数的真值为不知道的数值 θ^*
- 在频率学派观点中，该估计的不确定性通过计算其抽样分布得到（参数为一个值，而非随机变量，但 $\hat{\theta}$ 是随机变量）
  - 假设从真实分布 $p\left(x|\theta^*\right)$ 进行S次抽样，每次的样本集的大小均为N, 得到数据集合

$$\mathcal{D}(s)=\left\{x_i^{(s)}\right\}_{i=1}^{N},\quad x_i\sim p\left(x|\theta^*\right)$$

  - 根据每次抽样得到的数据$\mathcal{D}(s)$，都会得到$\hat{\theta}(.)$一个估计 $\hat{\theta}=\hat{\theta}\left(\mathcal{D}(s)\right)$
  - 当 $S\to\infty$ 时，我们可以得到估计的抽样分布

Bootstrap

- 一种重采样技术（resampling）
  - 用Monte Carlo技术近似抽样分布
  - 与交叉验证类似

Bootstrap

- Bootstrap是一个很通用的工具，用来估计标准误差、置信区间和偏差。由Efron Bradley 于1979年提出，用于计算任意估计的标准误差
- 1980年代很流行，因为计算机被引入统计实践中来
  - Bagging、随机森林、GBDT等算法均采用了Bootstrap技术

基本思想

- 若我们知道参数的真值 θ^*，则我们可以产生S组模拟数据，每组数据的大小均为N，即

$$x_i^{(s)} \sim p\left(x \mid \theta^*\right),\ i = 1:N,\ s = 1:S$$

- 且我们可以根据第s组数据得到估计 $\theta^{(s)} = f\left(x_{1:N}^{(s)}\right)$
- 然后用经验分布去近似估计的抽样分布

- 问题：参数的真值 θ^* 未知
- 解决方案：
  - 参数Bootstrap：用 $\hat{\theta}$ 代替 θ^*，从分布 $p\left(x \mid \hat{\theta}\right)$ 产生样本
  - 非参数Bootstrap：从原始数据 $\mathcal{D} = \left(X_1, \ldots, X_N\right)$ 进行N次有放回采样N个数据，用经验分布近似真正的分布

非参数Bootstrap

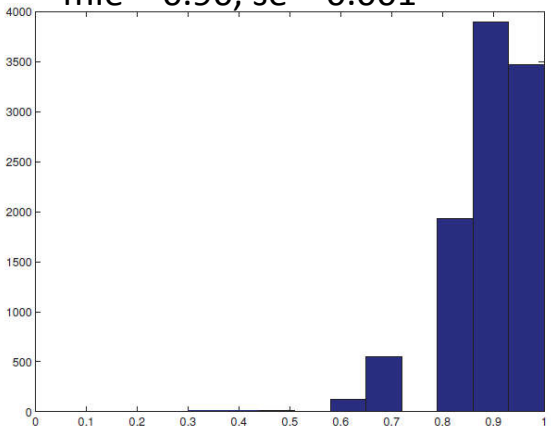
- 重复S次，
  - 1. 随机选择整数$i_1,\ldots,i_N$，每个整数的取值范围为$[1, N]$，选择每个$[1, N]$之间的整数的概率相等，均为
  - 2. 一组bootstrap样本为：$X^s=(X_{i1},\ldots,X_{iN})$

- Python函数：sklearn.utils.resample

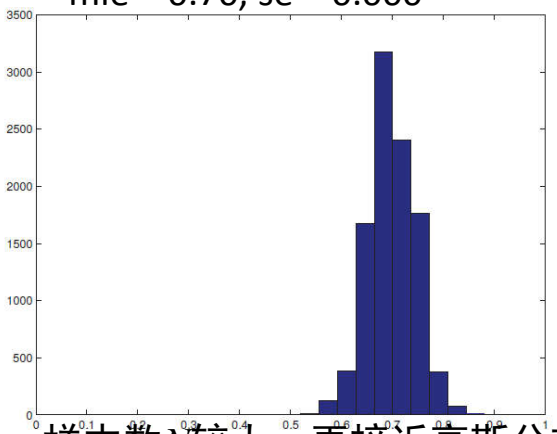
例：Bernoulli的抽样分布

- 对Bernoulli分布，用MLE估计其参数，然后采用参数Bootstrap方法抽样，得到如下抽样分布

Boot: true = 0.70, n=10,
mle = 0.90, se = 0.001



Boot: true = 0.70, n=100,
mle = 0.70, se = 0.000



样本数N较大，更接近高斯分布

► 课后练习

1、$X_1,...,X_N \sim \text{Unif}(0,\theta), \hat{\theta} = \max\left(X_1,...,X_N\right)$，计算该估计的偏差、标准误差和 MSE。

2、$X_1,...,X_N \sim \text{Unif}(0,\theta), \hat{\theta} = 2\overline{X}$，计算该估计的偏差、标准误差和 MSE。

- 3. 分别用Logistic回归和朴素贝叶斯分类器对Iris数据进行分类。

CSDN
不 止 于 代 码

THANK YOU

AI100