8.线性回归

课程目录

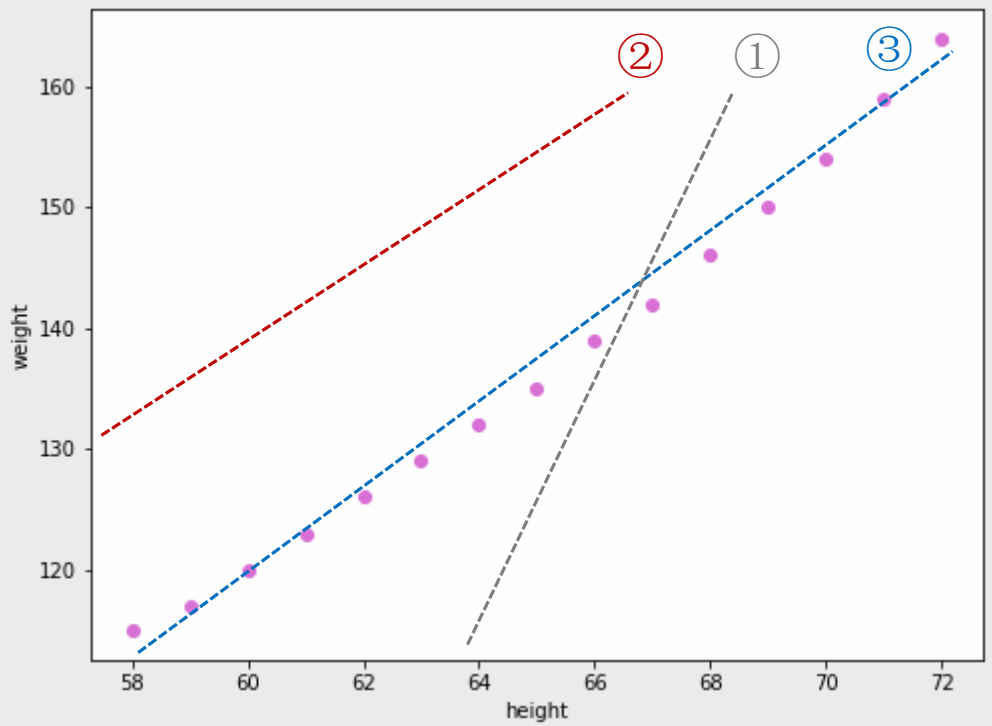
Course catalogue

本课目标

- 学习线性回归原理
- 学习回归诊断
- 学习岭回归与Lasso回归原理
- 学习如何解决过拟合与多重共线性问题
- 学习回归模型的Python实现

一元线性回归

height	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
weight	115	117	120	123	126	129	132	135	139	142	146	150	154	159	164



数据集：身高和体重

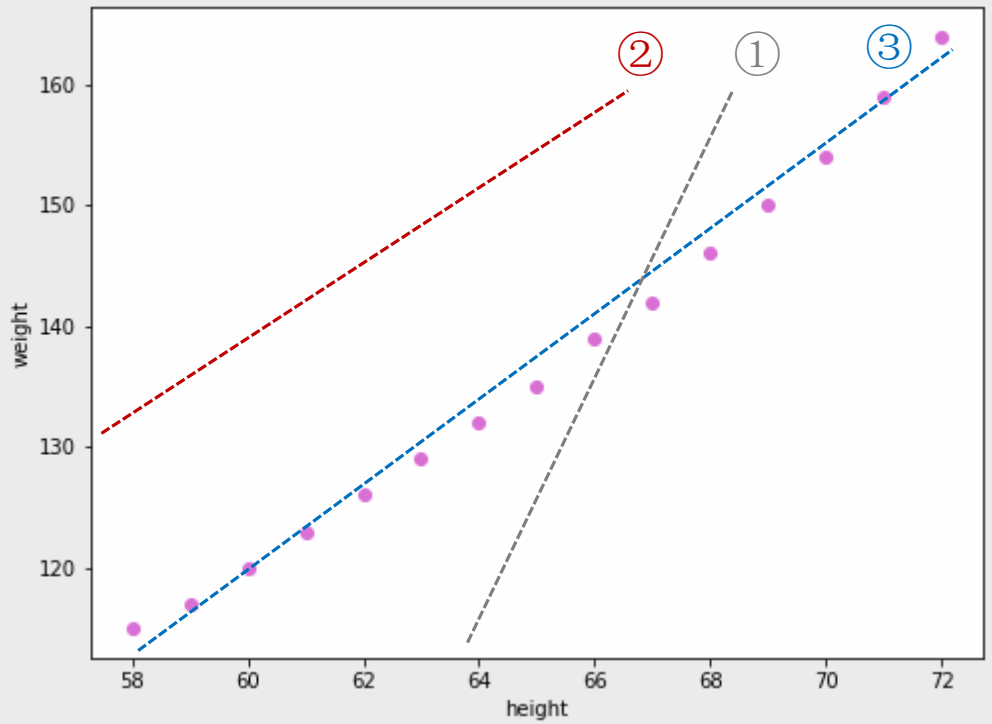
数据来自美国精算师协会早年的一项研究，包括身高（单位英寸）和体重（单位磅），共15个样本，分别代表了每个身高水平人群的平均体重。

根据散点图所展示的，身高和体重存在明显的相关关系，那么该如何表示这种关系？

【问题】应该选择哪条直线表示身高和体重之间的关系？为什么？

一元线性回归

height	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
weight	115	117	120	123	126	129	132	135	139	142	146	150	154	159	164



模型原理

【目的】找到一条直线，以尽可能准确的根据身高预测体重。该直线的表达式为：

$$y = wx + b$$

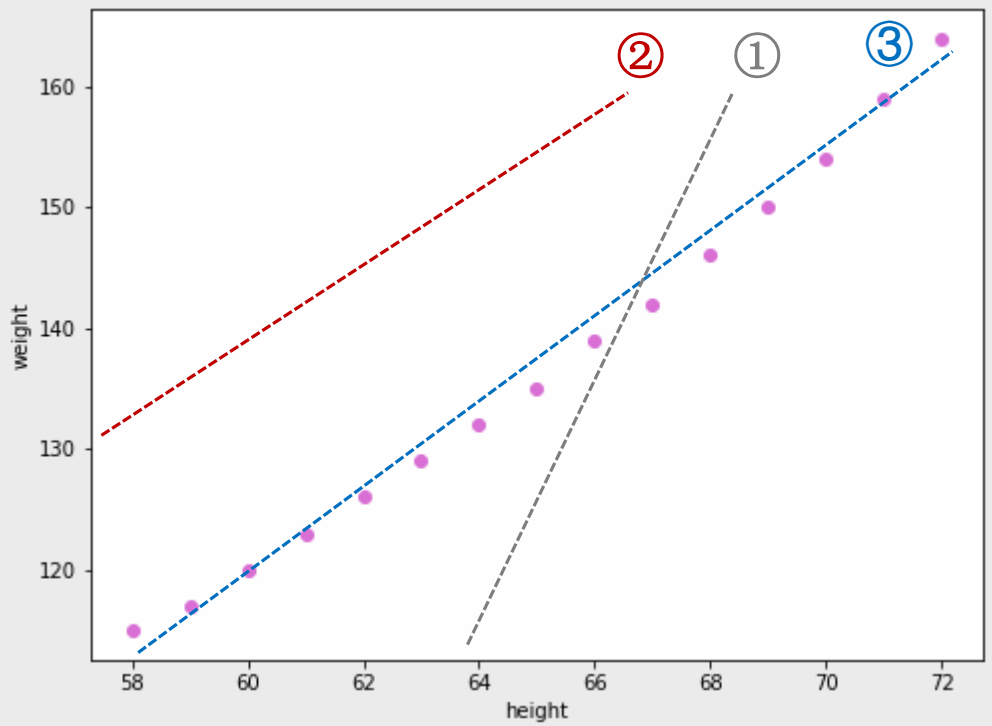
线性回归模型的目的是最小化均方误差，即

$$\min \sum_{i=0}^{m} (y_i - \hat{y}_i)^2$$

残差不是点到直线的距离！

一元线性回归

height	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
weight	115	117	120	123	126	129	132	135	139	142	146	150	154	159	164



评价指标

【模型最优解】

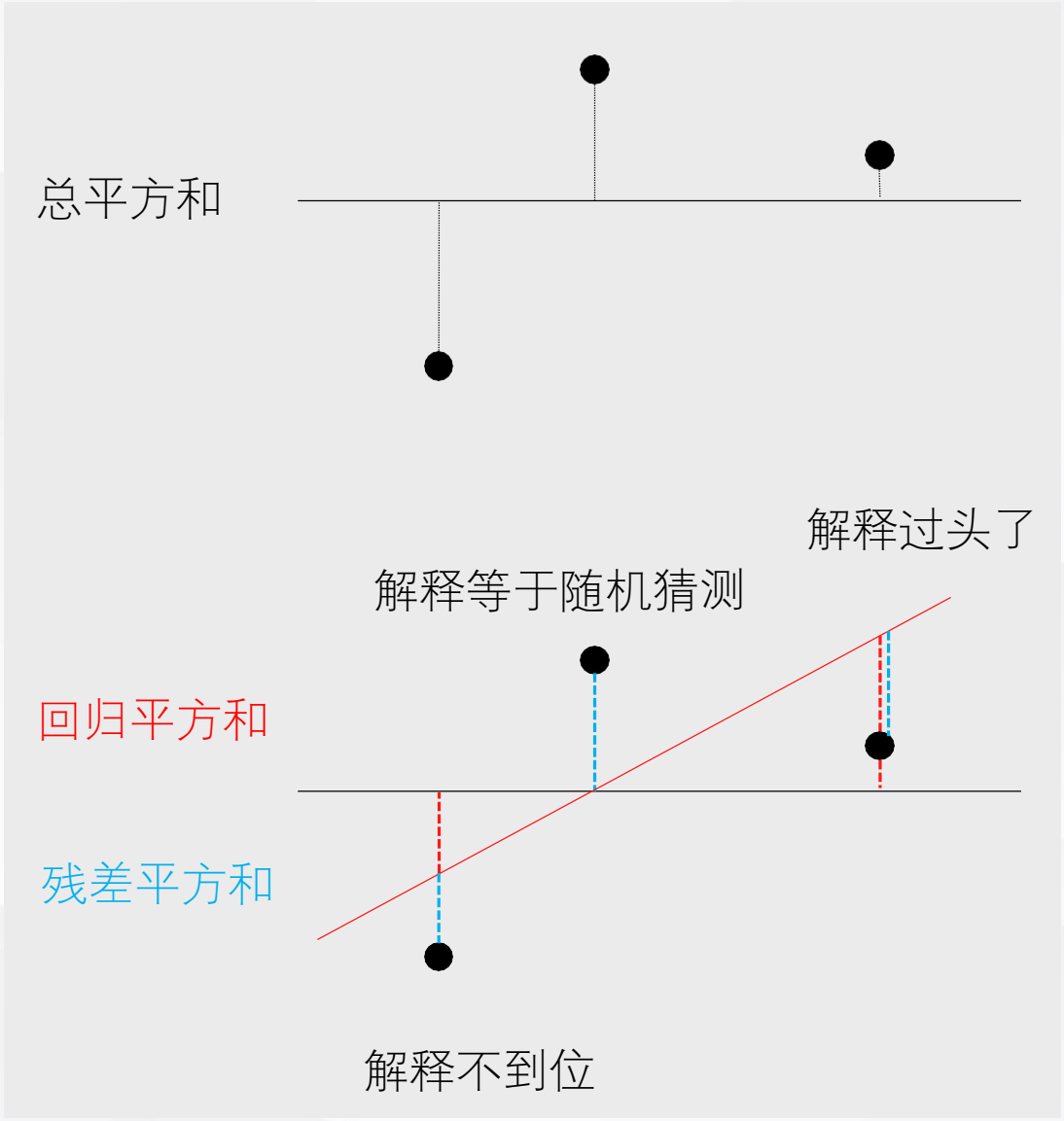
$$(w^*, b^*) = arg \min \sum_{i=0}^{m} (y_i - \hat{y}_i)^2$$

【评价指标】

均方误差：$$MSE = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

均方根误差：$RMSE = \sqrt{MSE}$

一元线性回归



R方与调整后的R方

- 回归平方和：$SSR = \sum(\hat{y}_i - \bar{y})^2$
- 残差平方和：$SSE = \sum(y_i - \hat{y}_i)^2$
- 总平方和：$SST = \sum_{i=0}^{m}(y_i - \bar{y})^2$
- $R^2 = 1 - \frac{SSE}{SST} = \frac{SSR}{SST}$
- $R_d^2 = 1 - \frac{m-1}{m-d-1}\left(\frac{SSR}{SST}\right)$

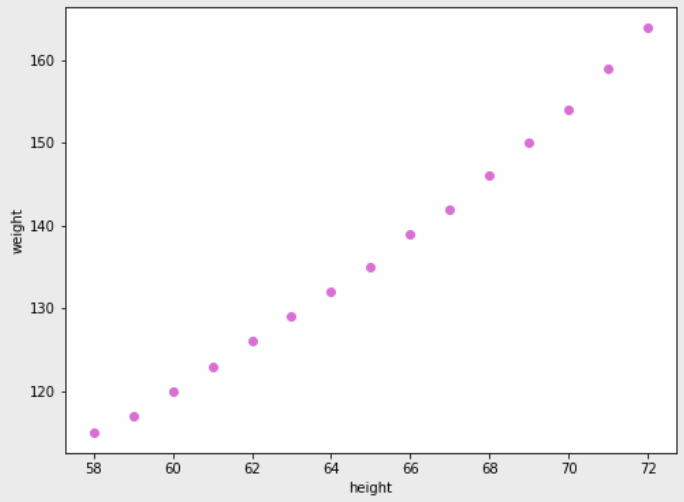
R平方表示模型的解释能力，但大小受自变量个数p的影响，随着自变量越多，R平方只会增加不会减少。

调整后R平方对变量数p进行惩罚，对结果进行改善。

一元线性回归

StatsModels
Statistics in Python

Scikit-Learn
ML in Python



Python解决方案

statsmodels和sklearn都可以实现线性回归。

statsmodels侧重于统计方法，比如最小二乘法、广义线性回归、方差分析等。

sklearn侧重于机器学习算法，比如回归、分类、聚类、降维等。

线性回归模型建议使用statsmodels实现，statsmodels操作方便，输出结果更直观友好。如果你熟悉R语言，就会发现statsmodels的线性回归输出结果，和R的输出结果十分相似。

一元线性回归

```python
# 读取数据
import pandas as pd
df = pd.read_csv('datas/women.csv')

# 数据可视化
import matplotlib.pyplot as plt
plt.scatter(df['height'],df['weight'],color='orchid')
plt.show()

# 数据预处理：增加截距项
import statsmodels.api as sm
df_const = sm.add_constant(df)
X = df_const.iloc[:,:-1]
y = df_const['weight']

# 建立OLS模型并预测
model = sm.OLS(y, X).fit()
y_pred = model.predict(X)

# 查看模型结果概要
model.summary()

# 绘制QQ图
sm.qqplot(model.resid,line='r')
plt.show()

# 残差分析
fig = plt.figure(figsize=(12,8))
fig = sm.graphics.plot_regress_exog(model, "height", fig=fig)
```

Python解决方案

读取csv文件：pandas.read_csv()

绘制散点图：matplotlib.pyplot.scatter()

增加截距项：statsmodels.api.add_constant()

建立OLS模型：model = sm.OLS(y, x).fit()

查看模型结果：model.summary()

模型预测：model.predict()

绘制QQ图：statsmodels.api.qqplot()

残差分析：statsmodels.api.graphics.plot_regress_exog()

一元线性回归

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.991
Model:	OLS	**Adj. R-squared:**	0.990
Method:	Least Squares	**F-statistic:**	1433.
Date:	Thu, 31 May 2018	**Prob (F-statistic):**	1.09e-14
Time:	16:18:55	**Log-Likelihood:**	-26.541
No. Observations:	15	**AIC:**	57.08
Df Residuals:	13	**BIC:**	58.50
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-87.5167	5.937	-14.741	0.000	-100.343	-74.691
height	3.4500	0.091	37.855	0.000	3.253	3.647

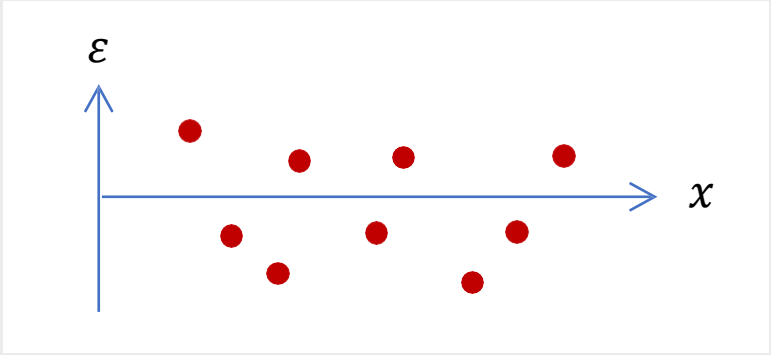
Omnibus:	2.396	**Durbin-Watson:**	0.315
Prob(Omnibus):	0.302	**Jarque-Bera (JB):**	1.660
Skew:	0.789	**Prob(JB):**	0.436
Kurtosis:	2.596	**Cond. No.**	982.

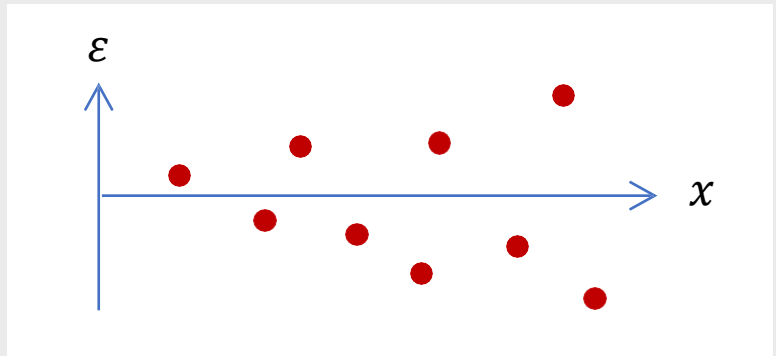
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

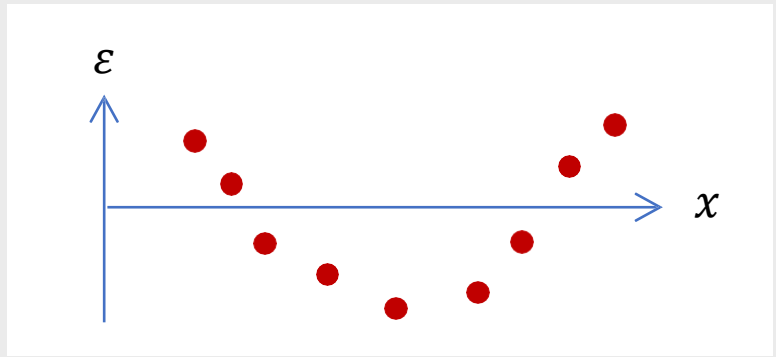
输出结果解读

- Prob (F-statistic): 模型显著性
- R-squared: 模型解释能力
- Adj. R-squared: 更稳健的模型解释能力
- Skew: 残差分布的偏度
- Kurtosis: 残差分布的峰度
- P>|t|: 自变量显著性
- coef: 自变量权重系数

一元线性回归







回归诊断：模型假设与残差分析

- 线性
  - 因变量和自变量之前存在线性关系。
- 独立性
  - 观测间独立 → 自相关 → 多见于时序数据
  - 自变量间独立 → 多重共线性
- 方差齐性（同方差）
  - 残差项ε_i的方差不依赖于自变量x_i的取值
- 正态性
  - 残差项ε_i服从正态分布

一元线性回归

“All models are wrong, but some are useful“

- George Box

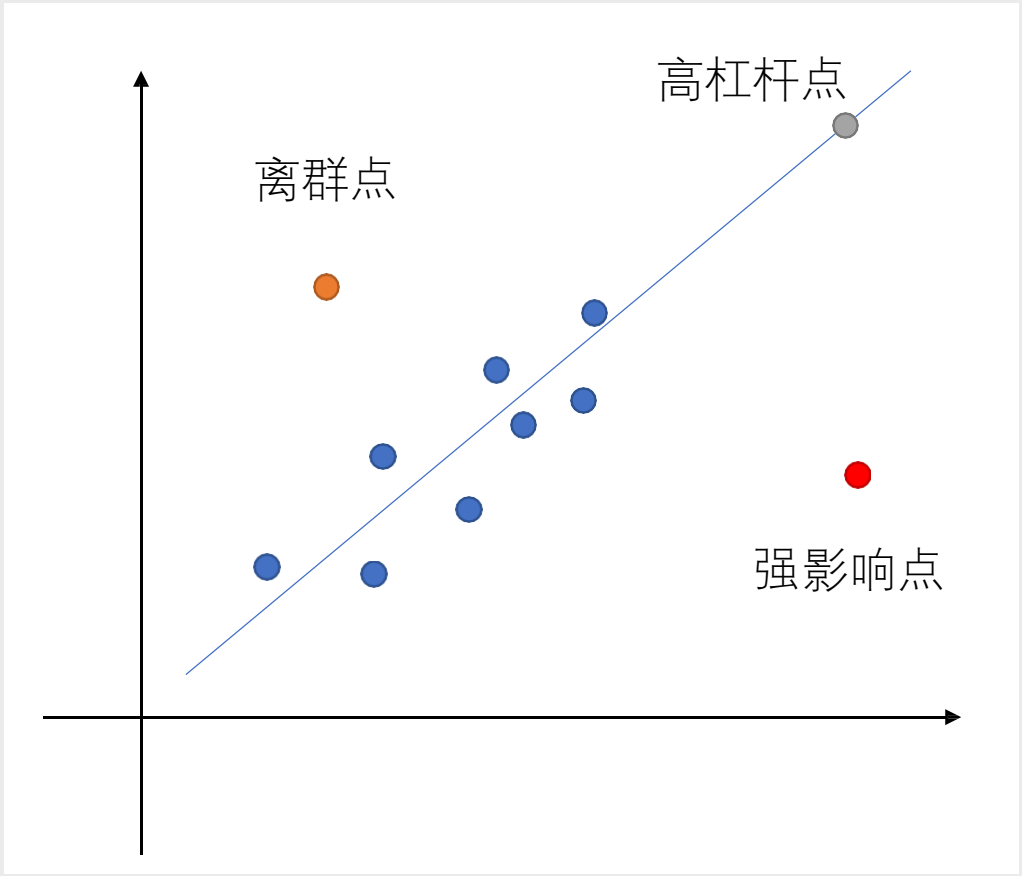
理论与应用的“冲突”

实际应用中，几乎没有任何一个数据能够完美地满足所有的理论假设，是不是意味着无法使用线性回归模型?

方差齐性：实际数据中，常方差假设往往遭到破坏，但由于中心极限定理的作用，只要样本足够大，那么统计推断方法依然是有效的。

正态性：实际中，数据很少满足正态性假设，但同样由于中心极限定理的作用，只要样本足够大，那么就不需要过于担心正态性假设。

一元线性回归

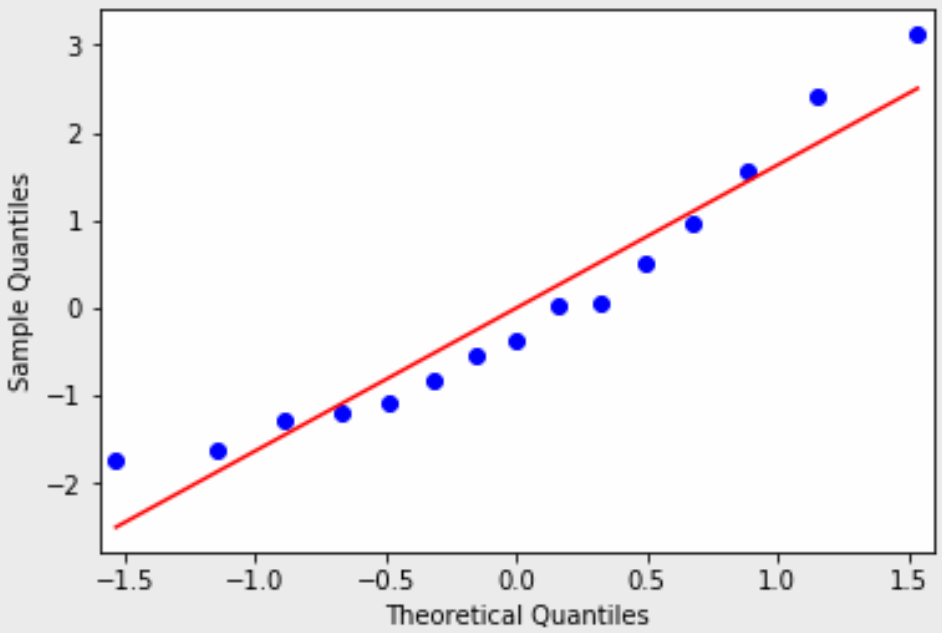


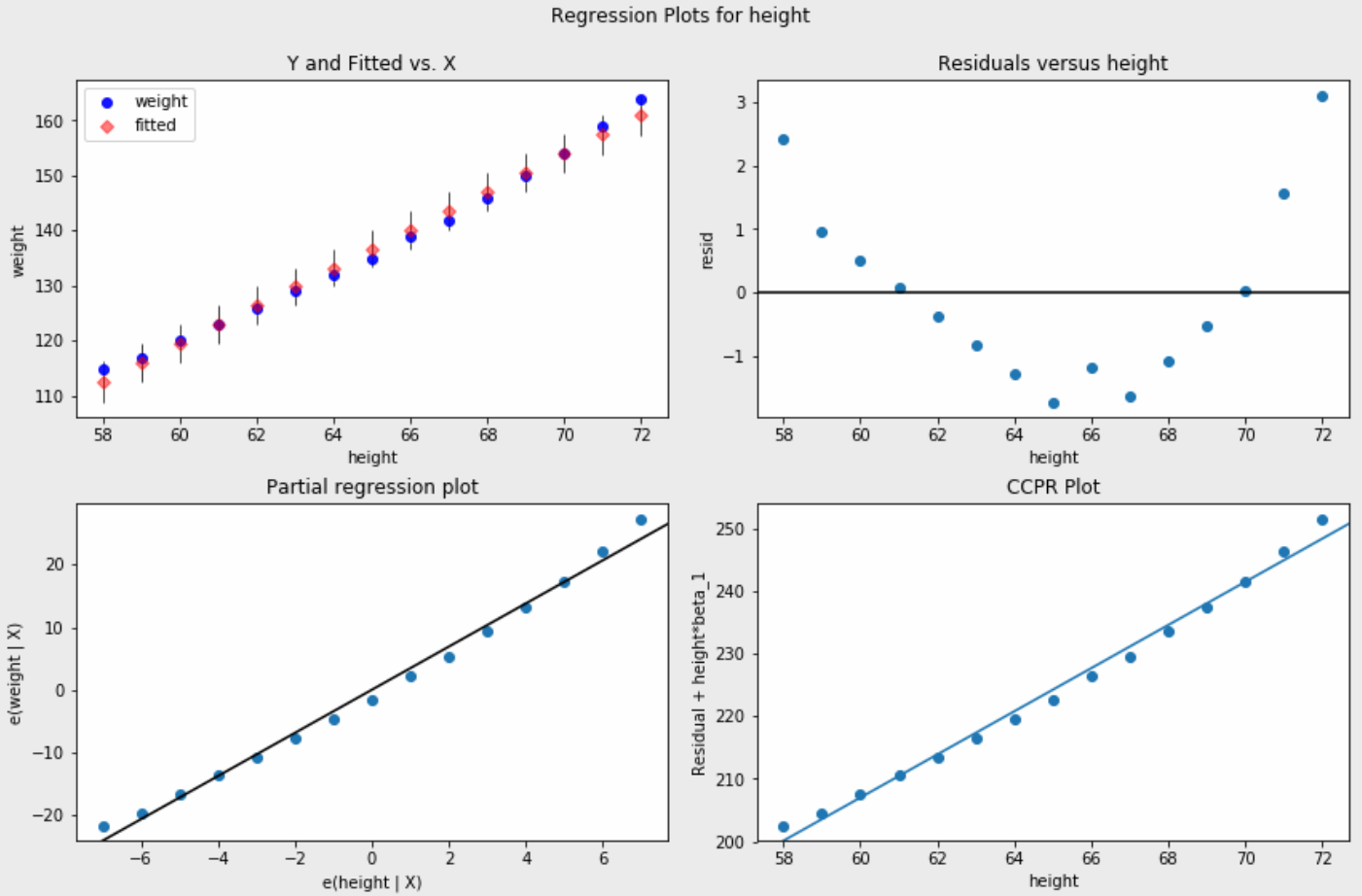
回归诊断：离群点、高杠杆点与强影响点

- 离群点：回归模型对其预测效果不佳（巨大的残差）。可以通过QQ图或标准化残差的绝对值进行判断。
- 高杠杆点：特征空间中的离群点。和因变量无关。可以通过帽子统计量进行判断。
- 强影响点：它对模型参数的估计产生的影响过大，非常不成比例。强影响点可以通过Cook距离即Cook’s D统计量来鉴别。
- 如果高杠杆点同时也是离群点，那么他是强影响点。

【思考】离群点、高杠杆点和强影响点如何影响模型？怎样处理他们？

一元线性回归



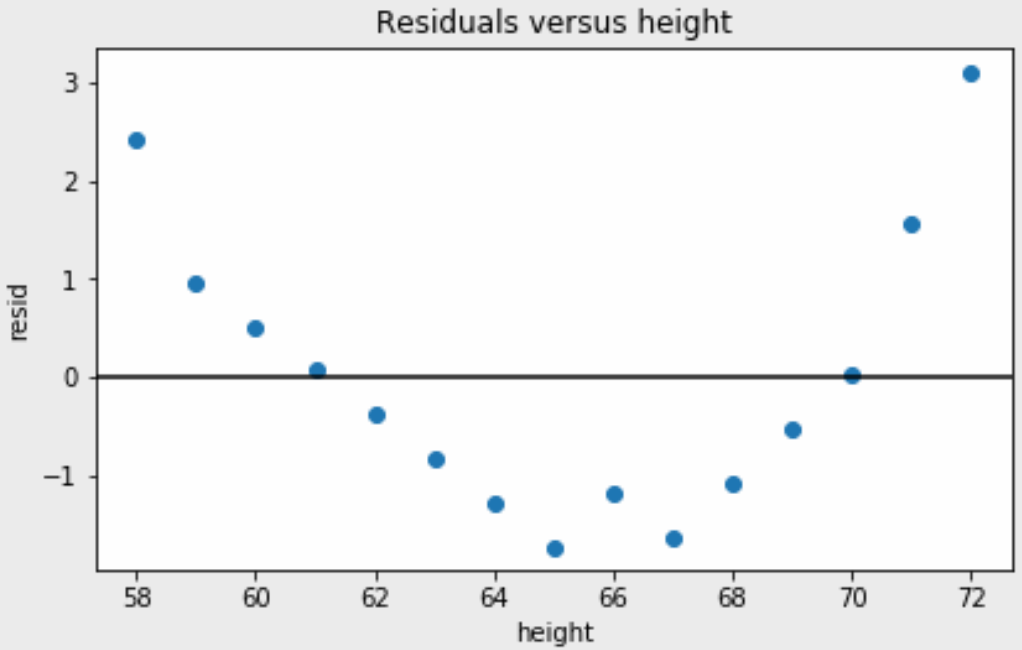


- QQ图有一定的弯曲，暗示残差可能不符合正态分布
- 残差分析发现残差没有均匀分布在水平线两侧，而是呈现一定“规律性”，暗示存在某些“规律”没有被发现
- 【问题】：违反了哪项回归模型假设？怎样改进？

课程目录

Course catalogue

多项式回归



一元线性回归模型改进方案

【未被发现的规律】

一条“抛物线”信息被隐藏在数据中，这意味着在模型中增加二次项也许能改善模型效果。

【改进方案】

$$y_{weight} = w_1 x_{height} + w_2 x^2{}_{heigh} + b$$

多项式回归

	const	height	weight
0	1.0	58	115
1	1.0	59	117
2	1.0	60	120
3	1.0	61	123
4	1.0	62	126

	const	height	weight	height_2
0	1.0	58	115	3364
1	1.0	59	117	3481
2	1.0	60	120	3600
3	1.0	61	123	3721
4	1.0	62	126	3844

改进方案代码实现

```
# 增加二次项
df_const['height_2'] = df_const['height']**2
# 定义自变量与因变量
X = df_const.loc[:,df_const.columns != 'weight']
y = df_const.loc[:,df_const.columns == 'weight’]
# 建立模型并查看
model = sm.OLS(y,X).fit()
model.summary()
```

多项式回归

OLS Regression Results

Dep. Variable:	weight	**R-squared:**	0.991
Model:	OLS	**Adj. R-squared:**	0.990
Method:	Least Squares	**F-statistic:**	1433.
Date:	Thu, 31 May 2018	**Prob (F-statistic):**	1.09e-14
Time:	16:18:55	**Log-Likelihood:**	-26.541
No. Observations:	15	**AIC:**	57.08
Df Residuals:	13	**BIC:**	58.50
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-87.5167	5.937	-14.741	0.000	-100.343	-74.691
height	3.4500	0.091	37.855	0.000	3.253	3.647

Omnibus:	2.396	**Durbin-Watson:**	0.315
Prob(Omnibus):	0.302	**Jarque-Bera (JB):**	1.660
Skew:	0.789	**Prob(JB):**	0.436
Kurtosis:	2.596	**Cond. No.**	982.

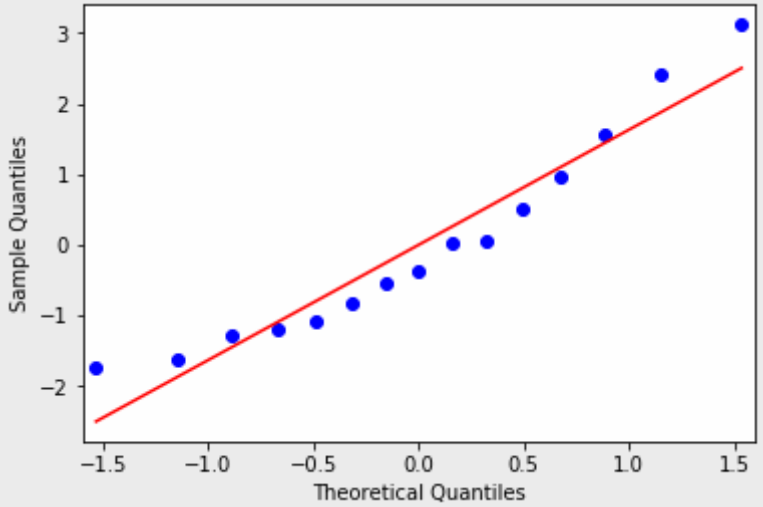
OLS Regression Results

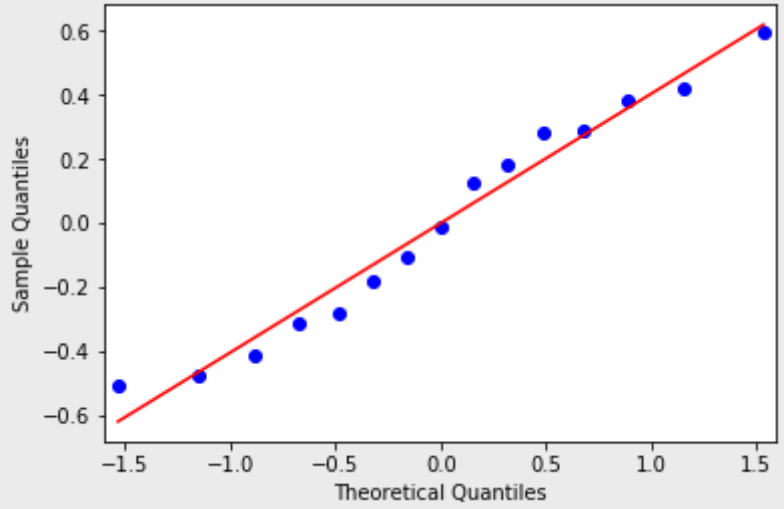
Dep. Variable:	weight	**R-squared:**	0.999
Model:	OLS	**Adj. R-squared:**	0.999
Method:	Least Squares	**F-statistic:**	1.139e+04
Date:	Fri, 01 Jun 2018	**Prob (F-statistic):**	2.13e-20
Time:	10:11:30	**Log-Likelihood:**	-5.2563
No. Observations:	15	**AIC:**	16.51
Df Residuals:	12	**BIC:**	18.64
Df Model:	2		
Covariance Type:	nonrobust		

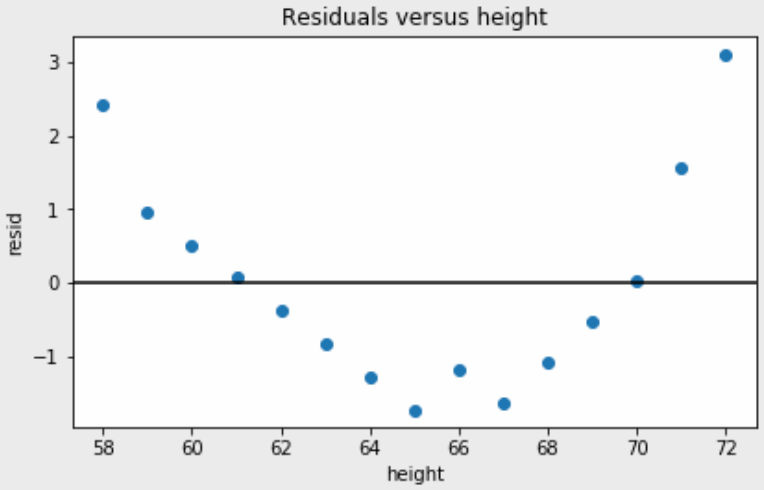
	coef	std err	t	P>|t|	[0.025	0.975]
const	261.8782	25.197	10.393	0.000	206.979	316.777
height	-7.3483	0.778	-9.449	0.000	-9.043	-5.654
height_2	0.0831	0.006	13.891	0.000	0.070	0.096

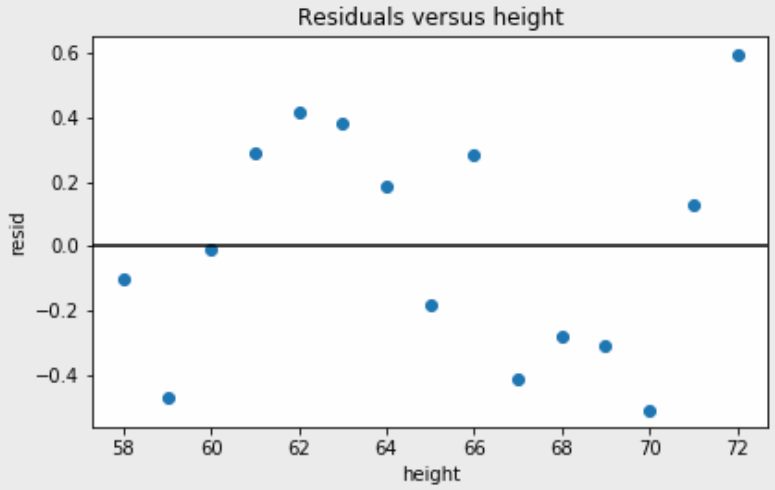
Omnibus:	2.449	**Durbin-Watson:**	1.144
Prob(Omnibus):	0.294	**Jarque-Bera (JB):**	1.033
Skew:	0.049	**Prob(JB):**	0.597
Kurtosis:	1.718	**Cond. No.**	1.09e+06

多项式回归

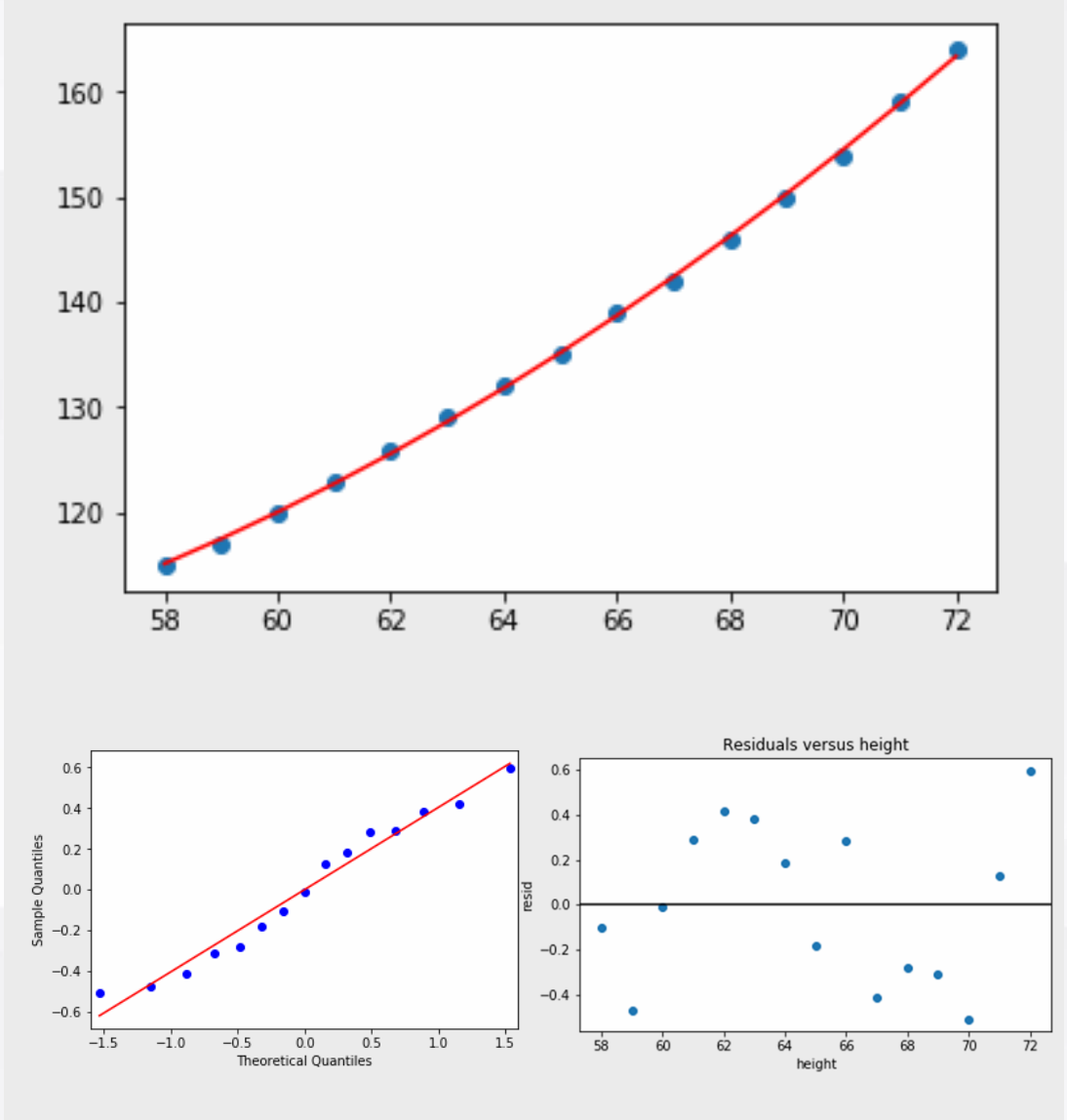

Sample Quantiles
Theoretical Quantiles
3
2
1
0
−1
−2
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Sample Quantiles
Theoretical Quantiles
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Residuals versus height
resid
height
3
2
1
0
−1
58
60
62
64
66
68
70
72


Residuals versus height
resid
height
0.6
0.4
0.2
0.0
−0.2
−0.4
58
60
62
64
66
68
70
72

多项式回归



问题

改进模型似乎也存在某种隐藏“规律”，似乎在模型中增加一个三次项会更好一些，但是这样太麻烦了，而且也并不是每一次都可以观测到这样明显的规律。

【问题】

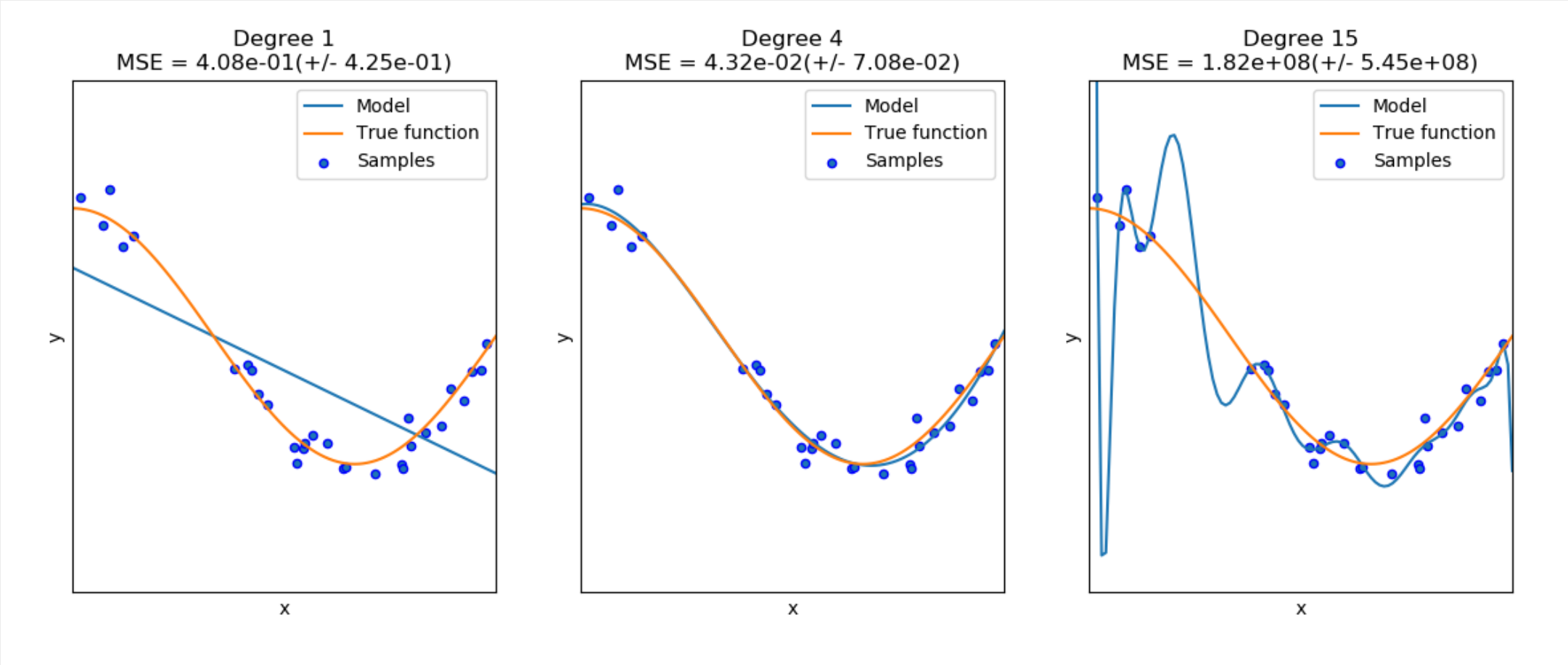
那么如果直接设置一个最高项很高很高的 n 项式，会不会更方便快捷一些？模型能够为我们自动筛选出恰当的最高项么？

【思考】

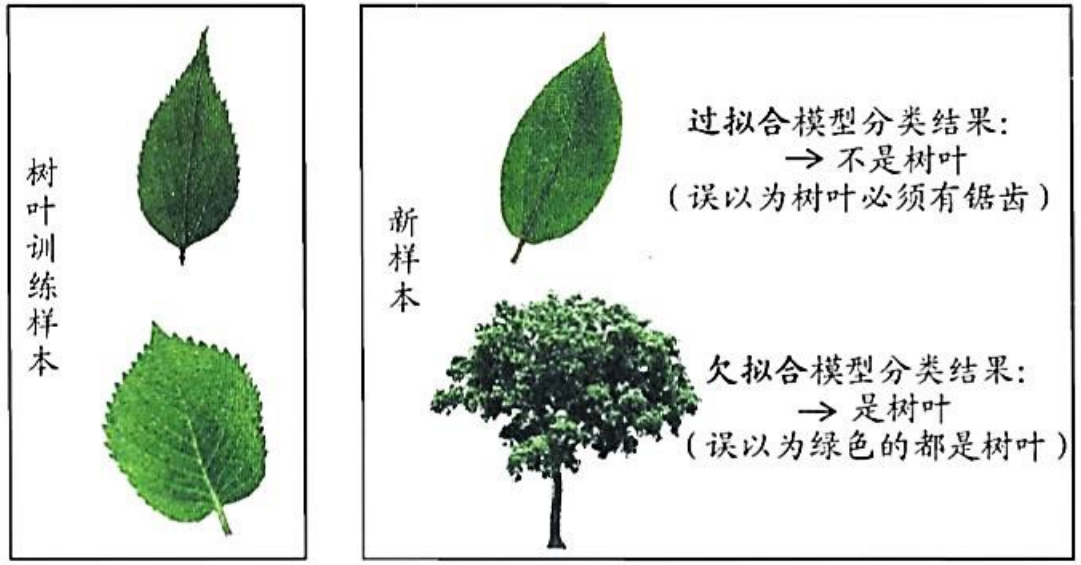
这样的处理方式会导致什么问题？

多项式回归

过拟合与欠拟合：



多项式回归

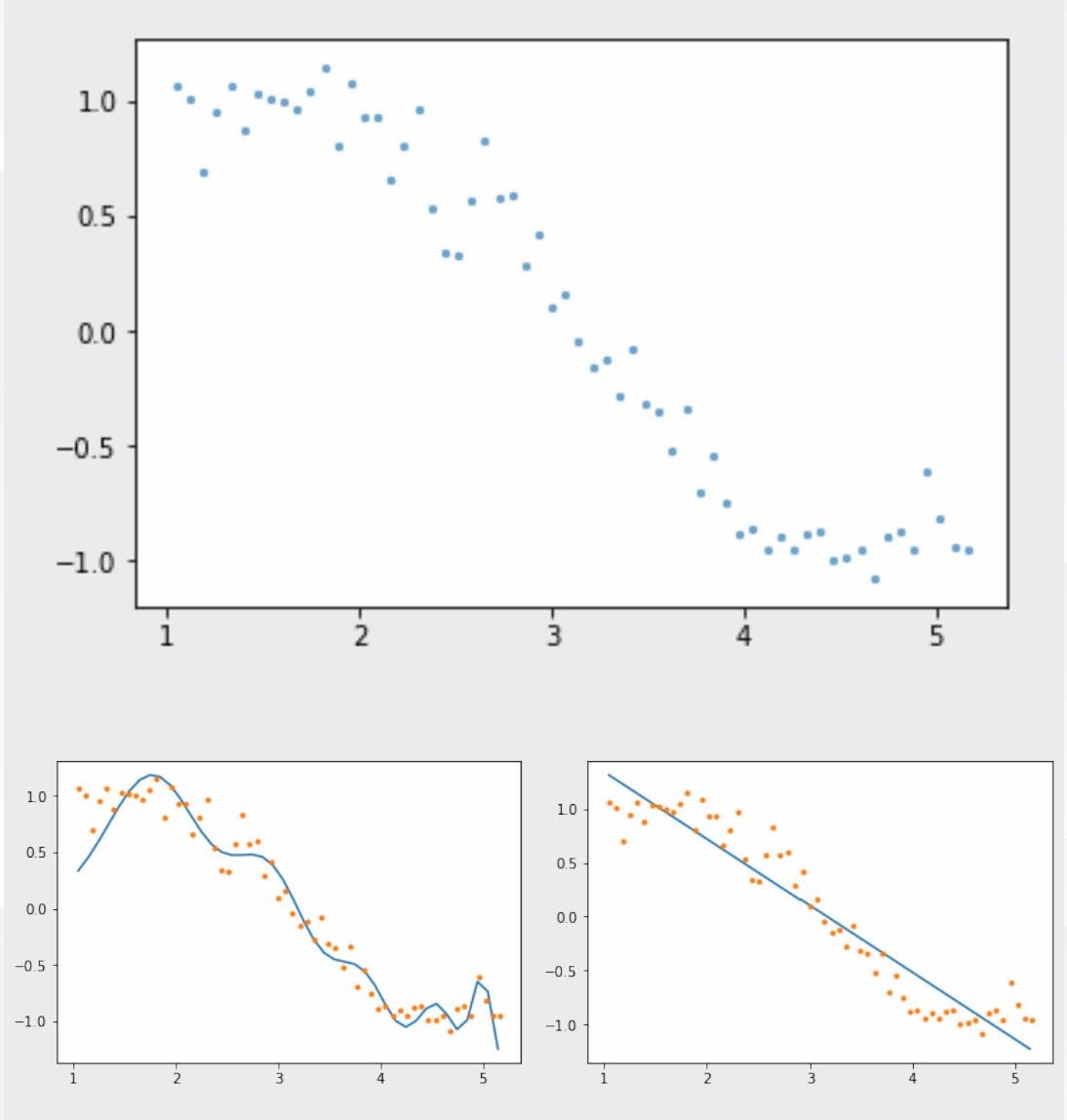


欠拟合与过拟合

学习器把训练样本学得"太好"了的时候，很可能已经把训练样本自身的一些特点当作了所有潜在样本都会具有的一般性质，这样就会导致泛化性能下降这种现象在机器学习中称为"过拟合"（overfitting）.与"过拟合"相对的是"欠拟合"（underfitting），这是指对训练样本的一般性质尚未学好.

- 过拟合：将随机误差拟合到模型。表现在训练集拟合特别好，测试机拟合很差。
- 欠拟合：没有拟合到足够的规律（数据信息）。表现在训练集和测试集拟合效果都很差。

多项式回归



欠拟合与过拟合案例分析

【Tips】如何生成多项式数据集

```
# 定义自变量和因变量
X = df['x'].values.reshape(-1,1)
y = df['y'].values.reshape(-1,1)
# 利用PolynomialFeatures建立多项式数据集
from sklearn.preprocessing import PolynomialFeatures
pf = PolynomialFeatures(22)
X_transform = pf.fit_transform(X)
```

课程目录

Course catalogue

多元线性回归

	mpg	hp	wt	drat	am
0	21.0	110	2.620	3.90	1
1	21.0	110	2.875	3.90	1
2	22.8	93	2.320	3.85	1
3	21.4	110	3.215	3.08	0
4	18.7	175	3.440	3.15	0
5	18.1	105	3.460	2.76	0
6	14.3	245	3.570	3.21	0
7	24.4	62	3.190	3.69	0
8	22.8	95	3.150	3.92	0
9	19.2	123	3.440	3.92	0

问题说明

数据集来自美国汽车杂志对汽车进行路试结果，共32个样本，包括四个字段，字段解释如下：

mpg Miles/(US) gallon

drat Rear axle ratio

wt Weight (1000 lbs)

hp Gross horsepower

am Transmission (0 = automatic, 1 = manual)

【问题】

如何依据其他变量，预测汽车油耗(mpg)?

多元线性回归

```
model = smf.ols(formula='mpg~hp+wt+drat+C(am)',data = df).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	mpg	**R-squared:**	0.843
Model:	OLS	**Adj. R-squared:**	0.820
Method:	Least Squares	**F-statistic:**	36.20
Date:	Mon, 04 Jun 2018	**Prob (F-statistic):**	1.75e-10
Time:	10:22:54	**Log-Likelihood:**	-72.769
No. Observations:	32	**AIC:**	155.5
Df Residuals:	27	**BIC:**	162.9
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	30.0271	6.185	4.855	0.000	17.336	42.718
C(am)[T.1]	1.5785	1.559	1.012	0.320	-1.621	4.778
hp	-0.0364	0.010	-3.706	0.001	-0.057	-0.016
wt	-2.7261	0.938	-2.907	0.007	-4.650	-0.802
drat	0.9810	1.377	0.712	0.482	-1.845	3.807

Omnibus:	3.503	**Durbin-Watson:**	1.615
Prob(Omnibus):	0.174	**Jarque-Bera (JB):**	2.981
Skew:	0.740	**Prob(JB):**	0.225
Kurtosis:	2.791	**Cond. No.**	2.26e+03

Python解决方案

一元调用：statsmodels.api.OLS

多元调用：statsmodels.formula.api.ols

语法：model = ols(formula='y~x1+x2+…',data=df)

离散变量：C(x3)

模型输出：model.summary()

注意事项：无需在数据集中添加截距项

离散变量解读【消失的 df.am 0】

多元线性回归

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	30.0271	6.185	4.855	0.000	17.336	42.718
C(am)[T.1]	1.5785	1.559	1.012	0.320	-1.621	4.778
hp	-0.0364	0.010	-3.706	0.001	-0.057	-0.016
wt	-2.7261	0.938	-2.907	0.007	-4.650	-0.802
drat	0.9810	1.377	0.712	0.482	-1.845	3.807

因为p-value < 0.05

↓

所以“拒绝”原假设

↓

所以选择备择假设

↓

所以保留显著项

↓

所以删掉非显著项

模型选择

【问题】

接下来是不是可以直接干掉不显著变量，只保留显著变量在模型中就可以了？

想一想，可不可以？为什么？

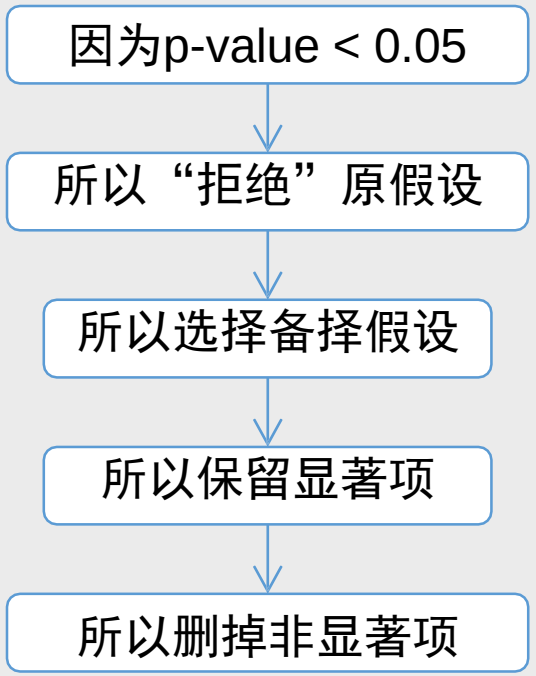
【tips】

结合假设检验的概念想一想

$$H_0: \beta_{drat} = 0 \text{ vs } H_0: \beta_{drat} \neq 0$$

多元线性回归

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	30.0271	6.185	4.855	0.000	17.336	42.718
C(am)[T.1]	1.5785	1.559	1.012	0.320	-1.621	4.778
hp	-0.0364	0.010	-3.706	0.001	-0.057	-0.016
wt	-2.7261	0.938	-2.907	0.007	-4.650	-0.802
drat	0.9810	1.377	0.712	0.482	-1.845	3.807



模型选择

在假设检验中，“拒绝”的反义词不是“接受”，而是“不能拒绝”，更严谨的说法，叫做“没有证据证明”。

以drat为例，显著性检验结果表明，“不能拒绝”或者“没有证据表明”$\beta_{drat} = 0$，即假设检验结果只能告诉我们，hp 和 wt 对模型非常重要，但无法排除其他因变量也有预测能力的可能。

因此，将显著变量放入模型，干掉其他变量的做法，虽然可行，但并不是最好的办法。

多元线性回归

```
model = smf.ols(formula='mpg~hp+wt+drat+C(am)',data = df).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	mpg	**R-squared:**	0.843
Model:	OLS	**Adj. R-squared:**	0.820
Method:	Least Squares	**F-statistic:**	36.20
Date:	Mon, 04 Jun 2018	**Prob (F-statistic):**	1.75e-10
Time:	10:22:54	**Log-Likelihood:**	-72.769
No. Observations:	32	**AIC:**	155.5
Df Residuals:	27	**BIC:**	162.9

```
model = smf.ols(formula='mpg~hp+wt',data=df).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	mpg	**R-squared:**	0.827
Model:	OLS	**Adj. R-squared:**	0.815
Method:	Least Squares	**F-statistic:**	69.21
Date:	Mon, 04 Jun 2018	**Prob (F-statistic):**	9.11e-12
Time:	10:52:39	**Log-Likelihood:**	-74.326
No. Observations:	32	**AIC:**	154.7
Df Residuals:	29	**BIC:**	159.0

模型选择

AIC（Akaike Information Criterion，赤池信息准则）是日本统计学家赤池，根据极大似然估计原理，提出的一种常用的选择标准。AIC值较小的模型要优先选择，它说明模型用较少的参数获得了足够的拟合度。

BIC （Bayesian Information Criterion，贝叶斯信息准则）同样可以作为选择标准，使BIC达到最小的模型是“最优”模型。

多数情况下，AIC和BIC的结果大同小异，但结果不一致时，需注意BIC的惩罚项比AIC的力度要大，因此，AIC选出的模型更为保守（包含更多的变量），BIC恰恰相反。

多元线性回归

思考题：球队的胜率

赌博公司对某支球队的胜率进行预测，判断如下：

- 如果队员甲缺席比赛，球队的胜率将下降10%
- 如果队员乙缺席比赛，球队的胜率将下降5%

那么队员甲、乙同时缺席比赛，球队的胜率下降多少?

多元线性回归

```
model = smf.ols(formula='mpg~hp*wt',data=df).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	mpg	**R-squared:**	0.885
Model:	OLS	**Adj. R-squared:**	0.872
Method:	Least Squares	**F-statistic:**	71.66
Date:	Mon, 04 Jun 2018	**Prob (F-statistic):**	2.98e-13
Time:	14:12:59	**Log-Likelihood:**	-67.805
No. Observations:	32	**AIC:**	143.6
Df Residuals:	28	**BIC:**	149.5
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	49.8084	3.605	13.816	0.000	42.424	57.193
hp	-0.1201	0.025	-4.863	0.000	-0.171	-0.070
wt	-8.2166	1.270	-6.471	0.000	-10.818	-5.616
hp:wt	0.0278	0.007	3.753	0.001	0.013	0.043

交互项的Python实现

多元线性回归模型语法：

model = ols(formula='y~x1+x2+…',data=df)

多元线性回归模型交互项语法：

ols(formula='y~x1+x2+x1:x2',data=df)

语 法 简 写 ：

ols(formula='y~x1*x2',data=df)

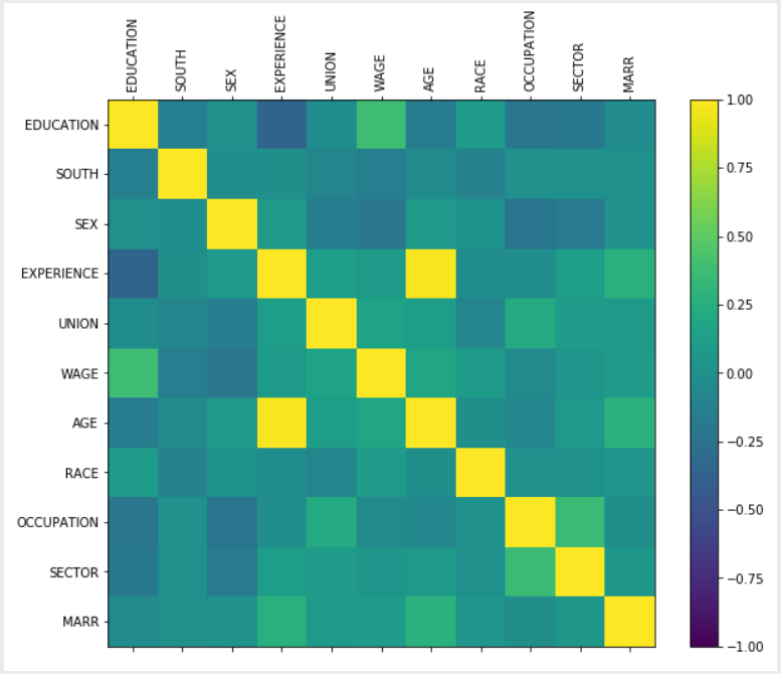
同理，x1*x2*x3等价于

$$x1 + x2 + x3 + x1{:}\,x2 + x1{:}\,x3 + x2{:}\,x3 + x1{:}\,x2{:}\,x3$$

多元线性回归

```
correlations = df.corr()
correlations
```

	EDUCATION	SOUTH	SEX	EXPERIENCE	UNION	WAGE	AGE	RACE	OCCUPATION	SECTOR	MARR
EDUCATION	1.000000	-0.140143	0.002031	-0.352676	-0.023886	0.381922	-0.150019	0.096016	-0.205998	-0.188853	-0.035522
SOUTH	-0.140143	1.000000	-0.021264	-0.007407	-0.086275	-0.141031	-0.038665	-0.114967	0.014740	0.000430	0.006523
SEX	0.002031	-0.021264	1.000000	0.075230	-0.157027	-0.205371	0.079179	0.026924	-0.220920	-0.170764	0.011225
EXPERIENCE	-0.352676	-0.007407	0.075230	1.000000	0.117926	0.087060	0.977961	-0.023990	-0.021779	0.111500	0.270900
UNION	-0.023886	-0.086275	-0.157027	0.117926	1.000000	0.161766	0.119466	-0.086544	0.229347	0.095849	0.093164
WAGE	0.381922	-0.141031	-0.205371	0.087060	0.161766	1.000000	0.176967	0.094708	-0.049069	0.045361	0.100579
AGE	-0.150019	-0.038665	0.079179	0.977961	0.119466	0.176967	1.000000	-0.004196	-0.069265	0.075918	0.278947
RACE	0.096016	-0.114967	0.026924	-0.023990	-0.086544	0.094708	-0.004196	1.000000	0.014694	0.001641	0.043644
OCCUPATION	-0.205998	0.014740	-0.220920	-0.021779	0.229347	-0.049069	-0.069265	0.014694	1.000000	0.364584	-0.011384
SECTOR	-0.188853	0.000430	-0.170764	0.111500	0.095849	0.045361	0.075918	0.001641	0.364584	1.000000	0.056050
MARR	-0.035522	0.006523	0.011225	0.270900	0.093164	0.100579	0.278947	0.043644	-0.011384	0.056050	1.000000



多重共线性

多重共线性是指变量之间存在高度相关关系。可以通过相关系数矩阵和方差膨胀因子（VIF）判断。Python中可以使用以下代码查看：

相关系数矩阵：df.corr()

方差膨胀因子：statsmodels.stats.outliers_influence.variance_inflation_factor()

一般来说，VIF大于4，即认为存在多重共线性。

【思考】多重共线性违背哪个线性回归基本假设？

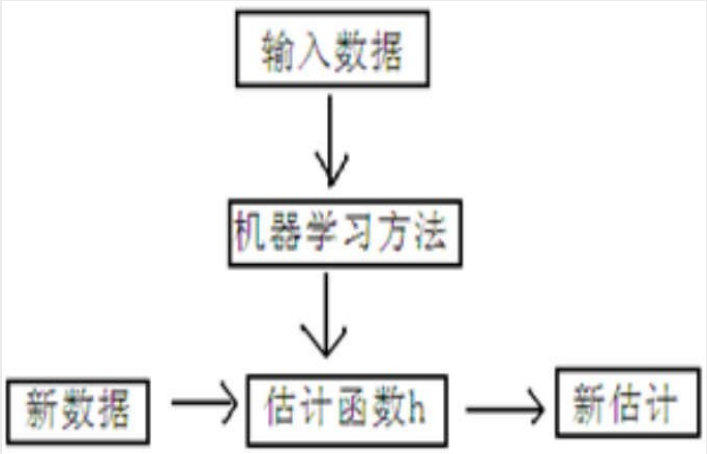
课程目录

Course catalogue

损失函数/代价函数求解

典型的线性回归模型

- 首先给出一个输入数据，我们的算法会通过一系列的过程得到一个估计的函数，这个函数有能力对没有见过的新数据给出一个新的估计，也被称为构建一个模型。就如同上面的线性回归函数。



线性回归 (linear regression)

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

目标：误差最小，如均方误差

$$\min \sum_{i=0}^{m} (y_i - \hat{y}_i)^2$$

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

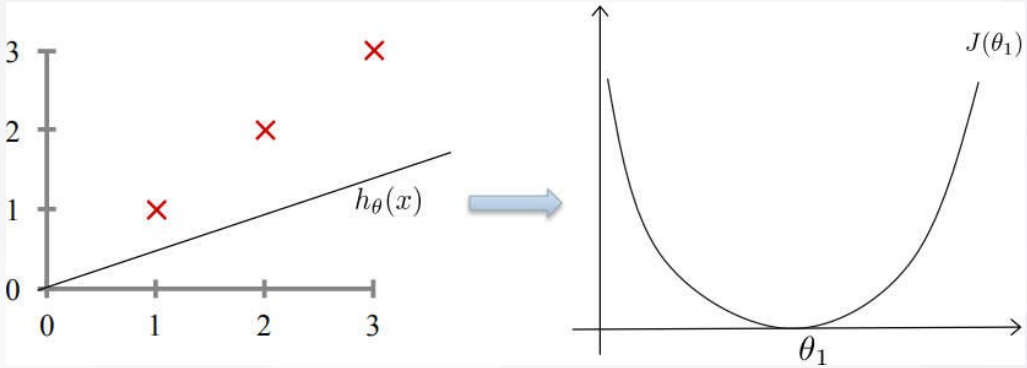
$$= \arg\min_{(w,b)} \sum_{i=1}^{m} (y_i - wx_i - b)^2$$

损失函数/代价函数求解

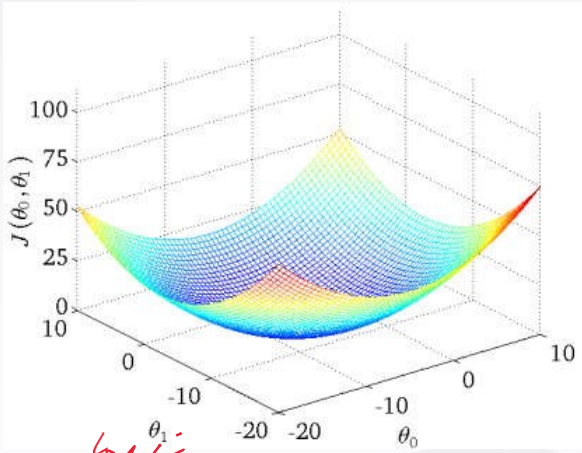
如何调整 参数 以使得代价函数 J(w)取得最小值有很多方 法。

其中有最小二乘法(min square)，是一种完全是数学描述的方法。
另一种常见的是梯度下降法。

最小化损失函数（loss function）-梯度下降法



二维



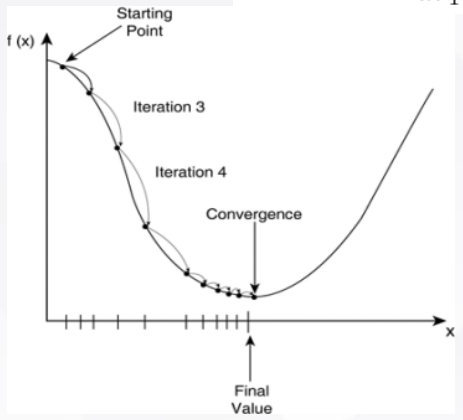
多维

选定线性回归模型后，只需要确定参数θ，就可以将模型用来预测。然而 θ 需要在 J(θ) 最小的情况下才能确定。因此问题归结为求极小值问题，使用梯度下降法。梯度下降法最大的问题是求得有可能是全局极小值，这与初始点的选取有关。

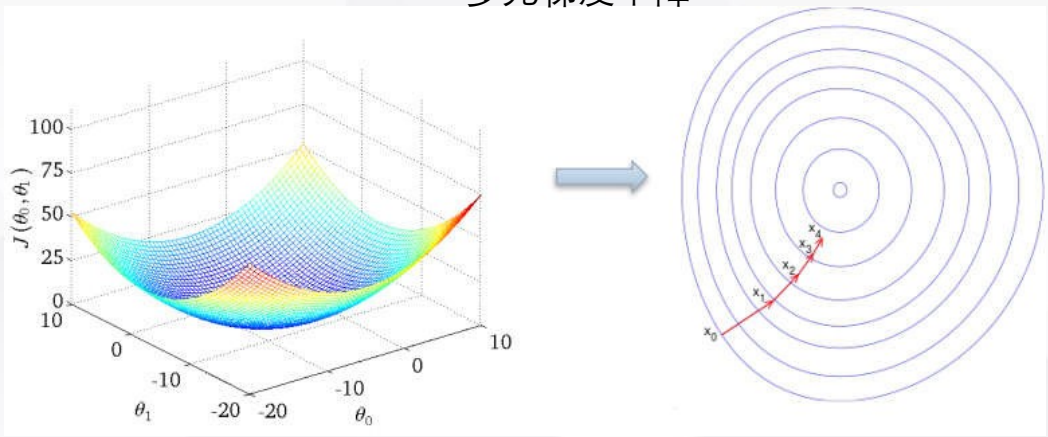
梯度下降

逐步最小化损失函数的过程
如同下山，找准方向(斜率)，每次迈进一小步，直至山底

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

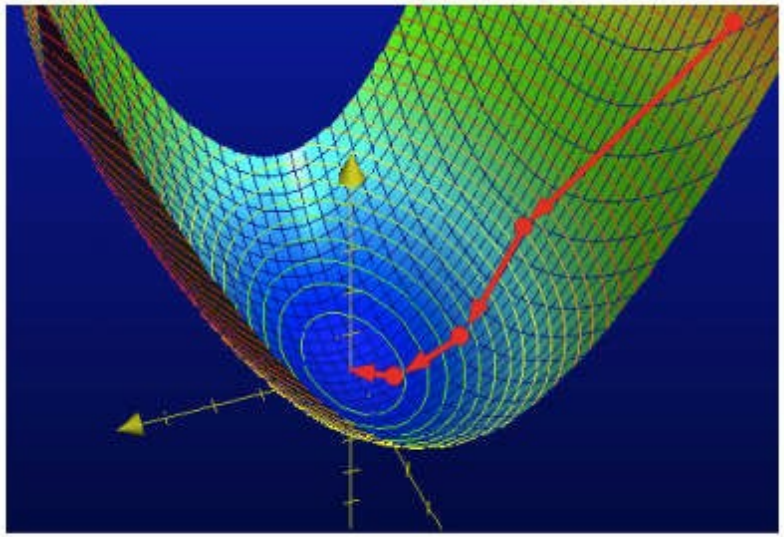


多元梯度下降



梯度下降法是按下面的流程进行的：
1）首先对 θ 赋值，这个值可以是随机的，也可以让θ 是一个全零的向量。
2）改变 θ 的值，使得 J(θ)按梯度下降的方向进行减少。

梯度下降与迭代



$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)}$$

}

多特征模式的梯度下降

- 梯度下降
  - 假如现在有n个特征/变量x_j(j=1…n)

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

}

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_2^{(i)}$$

$$\cdots$$

梯度下降

损失函数与梯度下降

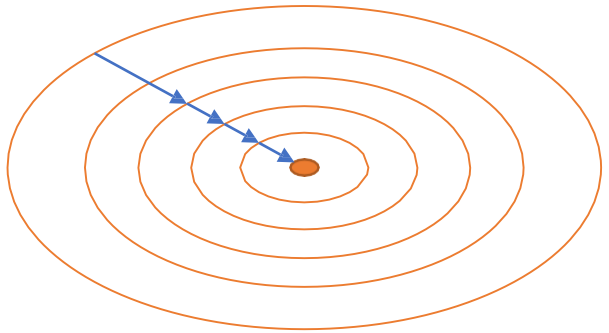
简单起见，我们考虑OLS的损失函数：

$$J(w,b)=\frac{1}{2m}\|y_i-\hat{y}_i\|_2{}^2$$

求最优解的一个思路，就是通过逐步逼近损失函数最小值，最终找到模型最优解，这就是梯度下降的基本思想。

想象一下，机器人站在一个随机初始位置，碗底代表损失函数最小值，机器人怎样才能走到碗底?

梯度下降



损失函数与梯度下降

对于机器人来说，这是一个迭代的过程，即：

Repeat:{

位置 = 位置+方向*步长

}

梯度下降正是通过不断逼近损失函数最小值的方式，求得系数的最优解。

我们可以利用等高线的方式，把梯度下降的过程还原出来。

损失函数/代价函数求解

如何调整 参数 以使得代价函数 J(w)取得最小值有很多方 法。

其中有最小二乘法(min square)，是一种完全是数学描述的方法。
另一种常见的是梯度下降法。

求解

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

最小化的过程，称为：线性回归模型的最小二乘“参数估计”—正规方程组方法

线性回归——最小二乘法的参数估计

分别对 w 和 b 求导：

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

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

线性回归——最小二乘法的参数估计

分别对 w 和 b 求导：

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

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

线性回归——最小二乘法的参数估计

分别对 w 和 b 求导：

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

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

上式令导数为 0：

$$0 = w\sum_{i=1}^{m}x_i^2 - \sum_{i=1}^{m}(y_i - b)x_i$$

$$w\sum_{i=1}^{m}x_i^2 = \sum_{i=1}^{m}y_i x_i - \sum_{i=1}^{m}bx_i$$

下式令导数为 0：

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

线性回归——最小二乘法的参数估计

$$0 = w\sum_{i=1}^{m} x_i^2 - \sum_{i=1}^{m}(y_i - b)x_i$$

$$w\sum_{i=1}^{m} x_i^2 = \sum_{i=1}^{m} y_i x_i - \sum_{i=1}^{m} b x_i$$

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

最终可以求得解析解：

$$w = \frac{\sum_{i=1}^{m} y_i\,(x_i - \bar{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2}$$

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

求得平方损失函数的极值点

线性回归——最小二乘法的参数估计

更一般的情况，样本有d个属性（多元线性回归），试图学得：

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

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

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

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

多元线性回归

同样采用最小二乘法求解，有

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

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

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

令其为零可得 $\hat{\boldsymbol{w}}$

然而，麻烦来了：涉及矩阵求逆！

- 若 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 满秩或正定，则 $\hat{\boldsymbol{w}}^* = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$
- 若 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 不满秩，则可解出多个 $\hat{\boldsymbol{w}}$

此时需求助于归纳偏好，或引入 正则化 (regularization) ⟶ 特征选择与支撑向量机