1 第一题

在平方误差情形下，样本内预测误差和训练误差：

$$Err_{in} = \frac{1}{N}\sum_{i=1}^{N} E_{Y^{new}} E_y \left(Y_i^{new} - \hat{f}(x_i)\right)^2$$

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} \left(y_i - \hat{f}(x_i)\right)^2$$

在每个表达式和展开式中增加和减去$f(x_i)$和$E\hat{f}(x_i)$，建立的训练误差中的乐观性为：

$$op = Err_{in} - E_y(\overline{err}) = \frac{2}{N}\sum_{i=1}^{N} cov(\hat{y}_i, y_i)$$

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i) + f(x_i) - E\hat{f}(x_i) + E\hat{f}(x_i) - \hat{f}(x_i))$$

$$= \frac{1}{N}[\sum(y_i - f(x_i))^2 + \sum(f(x_i) - E\hat{f}(x_i))^2 +$$

$$\sum(E\hat{f}(x_i) - \hat{f}(x_i)) + 2\sum(f(x_i) - E\hat{f}(x_i))(E\hat{f}(x_i) - \hat{f}(x_i))]$$

$$=: \frac{1}{N}(a_1 + b + c + d_1 + e_1 + f)$$

同理：$\frac{1}{N}\sum_{i=1}^{N}(Y_i^{new} - \hat{f}(x_i))^2 = \frac{1}{N}(a_2 + b + c + d_2 + e_2 + f)$

其中：$a_2 = \sum E_{Y^{new}}(Y_i^{new} - f(x_i))^2, d_2 = 2$

$\sum E_{Y^{new}}(Y_i^{new} - f(x_i))(f(x_i) - E\hat{f}(x_i))$

$e_2 = 2\sum E_{Y^{new}}(Y_i^{new} - f(x_i))(E\hat{f}(x_i) - \hat{f}(x_i))$

抱歉，下面因为能力不够，没有做出来

2 第二题

Prostate.data 数据

执行最佳子集线性回归分析，计算预测误差的 AIC\BIC\5 折交叉验证和 10 折交叉验证，632 估计，并讨论结果：

选择线性回归变量子集的直接方法是尝试所有可能的组合，并选择一个最小化某些标准的组合。这就是最佳子集回归的目标。对于每个k∈{1,2，...，p}，其中p是可用特征的总数，它选择大小为k的子集，其给出最小的残差平方和。然而，平方和不能用作确定k本身的标准，因为它必然随k减小：模型中包含的变量越多，其残差越小。但这并不能保证更好的预测性能。这就是为什么应该使用另一个标准来选择最终模型的原因。对于专注于预测的模型，测试数据上的（可能是交叉验证的）错误是常见的选择。

In [70]:

```
#读取数据
import pandas as pd
import numpy as np
import itertools
from sklearn.linear_model import LinearRegression
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
import sklearn
# K折交叉验证模块
from sklearn.model_selection import cross_val_score
import warnings
warnings.filterwarnings("ignore")

data = pd.read_csv('../data/prostate.txt',
                   header=0,sep='\t',encoding='utf-8',index_col=0)
X = np.array(data.iloc[:,[i for i in range(0,8)]])
Y = np.array(data['lpsa'])

# X_train = np.array(data[data['train']=='T'].iloc[:,[i for i in range(0,8)]])
# X_test = np.array(data[data['train']=='F'].iloc[:,[i for i in range(0,8)]])
# Y_train = np.array(data[data['train']=='T']['lpsa'])
# Y_test = np.array(data[data['train']=='F']['lpsa'])
```

包含K折交叉验证的最佳子集回归，验证统计量包括：AIC，BIC，平均绝对误差

In [76]:

```
def K_folds_regression(X,Y,K):
    results = pd.DataFrame(columns=['num_features', 'features', 'MAE'])
#     ,'AIC','BIC'
    for k in range(1, X.shape[1] + 1):
        # Loop over all possible subsets of size k
        #C_shape(1)^k,k是选k个自变量，不是K折
        for subset in itertools.combinations(range(X.shape[1]), k):
            subset = list(subset)
            #循环建模所有子集（2^p-1个）
            linreg_model = LinearRegression(normalize=True)#.fit(X_train[:,subset],
            #k折交叉验证
            mae_scores = cross_val_score(linreg_model,X[:, subset],Y,cv=K, scoring='
#             linreg_prediction = linreg_model.predict(X_test[:, subset])
#             #计算测试集平均绝对误差
#             linreg_mae = np.mean(np.abs(Y_test - linreg_prediction))
            results = results.append(pd.DataFrame([{'num_features': k,
                                                    'features': subset,
                                                   'MAE': -mae_scores.mean()}]))
    # Inspect best combinations
    results = results.sort_values('MAE').reset_index(drop=True)
    print(results)
    # MAE效果最好的模型
    best_subset_model = LinearRegression(normalize=True).fit(X[:, results['features'
    best_subset_coefs = dict(
        zip(['Intercept'] + data.columns.tolist()[:-1],
            np.round(np.concatenate((best_subset_model.intercept_, best_subset_model
    )
    print('Best Subset Regression MAE: {}'.format(np.round(results['MAE'][0], 3)))
    print('Best Subset Regression coefficients:')
    print(best_subset_coefs)
```

In [77]:

```
K=5
K_folds_regression(X,Y,K)
```

```
    num_features          features       MAE
0              3         [0, 1, 7]  0.738107
1              4      [0, 1, 2, 7]  0.740253
2              3         [0, 1, 2]  0.745129
3              5   [0, 1, 4, 5, 7]  0.745662
4              2            [0, 1]  0.745673
..           ...               ...       ...
250            3         [2, 3, 7]  1.103974
251            1               [2]  1.108408
252            4      [2, 3, 6, 7]  1.113667
253            1               [3]  1.121181
254            2            [2, 3]  1.126282

[255 rows x 3 columns]
Best Subset Regression MAE: 0.738
Best Subset Regression coefficients:
{'Intercept': -0.889, 'lcavol': 0.595, 'lweight': 0.67, 'age': 0.005}
```

使用5折交叉验证，选择平均MAE最小的模型，它的值为0.738。最终全部数据建模，最终得到的模型及参数值如下：

$$lpsa = -0.889 + 0.595lcavol + 0.67lweight + 0.005age$$

In [79]:

```
K=10
K_folds_regression(X,Y,K)
```

```
    num_features                features       MAE
0              5        [0, 1, 2, 3, 7]  0.656027
1              4           [0, 1, 2, 7]  0.659407
2              7  [0, 1, 2, 3, 4, 5, 6]  0.659766
3              6     [0, 1, 2, 3, 5, 7]  0.661299
4              7  [0, 1, 2, 3, 4, 5, 7]   0.66187
..           ...                     ...       ...
250            2                  [2, 6]  0.955417
251            3              [1, 2, 3]  0.963738
252            1                   [3]  0.990783
253            1                   [2]   0.99333
254            2                  [2, 3]  0.993953

[255 rows x 3 columns]
Best Subset Regression MAE: 0.656
Best Subset Regression coefficients:
{'Intercept': 0.35, 'lcavol': 0.612, 'lweight': 0.67, 'age': -0.02, 'l
bph': 0.069, 'svi': 0.006}
```

使用20折交叉验证，选择平均MAE最小的模型，它的值为0.656。最终全部数据建模，最终得到的模型及参数值如下：

$$lpsa = -0.35 + 0.612lcavol + 0.67lweight - 0.02age + 0.069lbph + 0.006svi$$

对AIC及BIC的值进行交叉验证同理，这里不再展开。

3 第三题

推导一个给定的观测被包含在一个自助法样本中的概率。假如从一个有 n 个观测的集合中得到了一个自助法样本。

(a) 第一个自助法观测不是原始样本中第 j 个观测的概率是多少？证明你的结论。

(b) 第二个自助法观测不是原始样本中第 j 个观测的概率是多少？

(c) 证明第 j 个观测不在自助法样本里的概率为 $(1-1/n)^n$。

(d) 当 $n=5$ 时，第 j 个观测在自助法样本里的概率是多少？

(e) 当 $n=100$ 时，第 j 个观测在自助法样本里的概率是多少？

(f) 当 $n=10\,000$ 时，第 j 个观测在自助法样本里的概率是多少？

(g) 作图展示，对于 n 从 1 到 100 000 的每个整数，第 j 个观测在自助法样本里的概率。讨论观察到的结果。

(h) 现在研究一个样本量为 $n=100$ 的自助法样本包含第 j 个观测的概率。这里 $j=4$。首先反复地产生自助法样本，然后每次把第四个观测是否包含在自助法样本里记录下来。

```
> store=rep(NA, 10000)
> for(i in 1:10000){
    store[i]=sum(sample(1:100, rep=TRUE)==4)>0
}
> mean(store)
```

讨论得到的结果。

a. $\frac{n-1}{n}$

b. $\frac{n-1}{n}$

c. 证明：第j个观测不在自助法观测第k个样本里的概率均为$\frac{n-1}{n}$，根据乘法法则，第j个观测不在所有自助法样本里的概率为$(1-\frac{1}{n})^n$

d.$1-(1-\frac{1}{5})^5$

e. $1-(1-\frac{1}{100})^{100}$

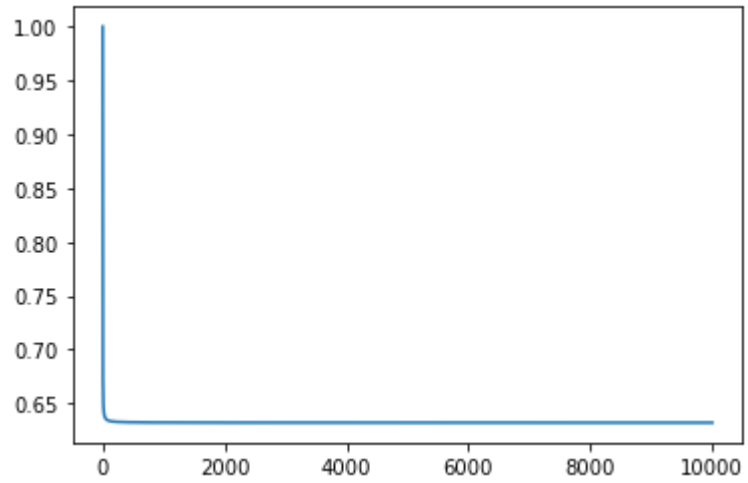
f. $1-(1-\frac{1}{10000})^{10000}$

In [92]:

```
#g题
from scipy.special import comb
import matplotlib.pyplot as plt
numbers = np.arange(1,10001,1)
scores = []
for i in numbers:
    scores.append(1-(1-1/i)**i)
plt.plot(numbers,scores)
```

Out[92]:

```
[<matplotlib.lines.Line2D at 0x17f23a040>]
```



可见当n=1的时候，概率为1，之后慢慢收敛为0.632左右。

In [108]:

```
#h题
import random
number_set = np.arange(1,101).tolist()
count = 0
for i in range(100000):
    results = random.choices(population=number_set, k=100)
    if 4 in results:
        count+=1
print(count/100000)
```

```
0.63659
```

经过100000次的蒙特卡洛模拟，最终求得在n=100的自助法样本中，包含第4个观测的概率大约为0.637。

4 第四题

在一个模拟数据集上使用交叉验证法。

(a) 生成一个模拟数据集如下：

```
> set.seed(1)
> y=rnorm(100)
> x=rnorm(100)
> y=x-2*x^2+rnorm(100)
```

在这个数据集中，n 和 p 分别是多少？用方程的形式写出生成这个数据的模型。

(b) 作 X 对 Y 的散点图。讨论结果。

(c) 设置一个随机种子数，然后计算用最小二乘法来拟合下面四个模型所产生的 LOOCV 误差：

i. $Y=\beta_0+\beta_1X+\varepsilon$

ii. $Y=\beta_0+\beta_1X+\beta_2X^2+\varepsilon$

iii. $Y=\beta_0+\beta_1X+\beta_2X^2+\beta_3X^3+\varepsilon$

iv. $Y=\beta_0+\beta_1X+\beta_2X^2+\beta_3X^3+\beta_4X^4+\varepsilon$

注意，可以用 data.frame() 函数来创建一个同时包含 X 和 Y 的数据集。

(d) 用另外一个随机种子来重复步骤 (c)，并讨论结果。结果跟步骤 (c) 中所得到的结果一样吗？为什么？

(e) 步骤 (c) 中的哪个模型有最小的 LOOCV 误差？这跟你预计的结果一样吗？解释你的结论。

(f) 讨论用最小二乘法拟合 (c) 中的每个模型所得到的系数估计的统计意义。这些结果与用交叉验证法所得到的结论一致吗？

a

In [116]:

```
#随机数种子
import random
random.seed(1)
import numpy as np
np.random.seed(1)

x = np.random.normal(0,1,100)
y = x-2*x**2+np.random.normal(0,1,100)
```

n=100,p=1,模型如下：

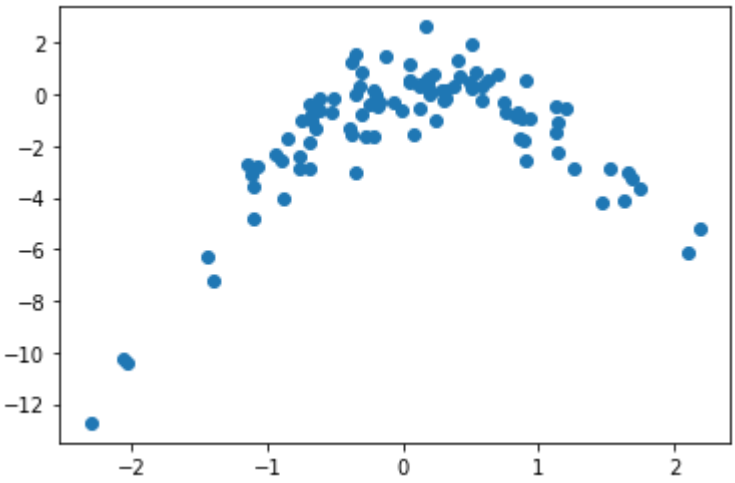
$$y = x - 2x^2 + \epsilon$$

In [119]:

```
plt.scatter(x,y)
```

Out[119]:

```
<matplotlib.collections.PathCollection at 0x281e146a0>
```



x对于y的散点图和一个标准正态分布的概率密度函数相似

5 第五题

考虑 MASS 程序包中的 Boston 住房数据集。

(a) 基于这个数据集，给出一个对 medv（房价中位数）的总体均值的估计，记为 $\hat{\mu}$。

(b) 给出一个对 $\hat{\mu}$ 的标准误差的估计。解释这个结果。
提示：可以用样本的标准差除以观测数的平方根来计算样本均值的标准误差。

(c) 用自助法来估计 $\hat{\mu}$ 的标准误差。这与在（b）中所得到的结果相比如何？

(d) 基于（c）中得到的自助法估计，给出对 medv 均值的 95% 置信区间。将这个置信区间与用 t.test(Boston$medv) 所得到的结果相比较。
提示：可以用公式 $[\hat{\mu}-2\mathrm{SE}(\hat{\mu}), \hat{\mu}+2\mathrm{SE}(\hat{\mu})]$ 来得到近似的 95% 的置信区间。

(e) 基于这个数据集，给出 medv 总体中位数的估计 $\hat{\mu}_{med}$。

(f) 现在想要估计 $\hat{\mu}_{med}$ 的标准误差。但并没有一个简单的公式来计算中位数的标准误差。不过，可以用自助法来估计中位数的标准误差。讨论计算的结果。

(g) 基于这个数据集，给出波士顿郊区的 medv 的 10% 分位数的估计，记为 $\hat{\mu}_{0.1}$。（这里可以使用 quantile() 函数。）

(h) 用自助法来估计 $\hat{\mu}_{0.1}$ 的标准误差。讨论结果。

In [ ]:

```
#a.
library(MASS)
summary(Boston)

set.seed(1)
attach(Boston)
medv.mean=mean(medv);medv.mean
#得22.533
```

In [ ]:

```
#b
Medv.err=sd(medv)/sqrt(length(medv));Medv.err
#得0.4088611
```

In [ ]:

```
#c
Boot.fn=function(data,index)return (mean(data[index]))
library(boot)
Bstrap=boot(medv,Boot.fn,1000);Bstrap
```

Bootstrap Statistics :

```
    original      bias    std. error
```

t1* 22.53281 0.005601581 0.4060352

In [ ]:

```
#d
t.test(medv)
c(bstrap$t0-2*0.4107,bstrap$t0+2*0.4107)
```

One Sample t-test

data: medv

t = 55.111, df = 505, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

21.72953 23.33608

sample estimates:

mean of x

22.53281

In [ ]:

```
#e
Medv.med=median(medv);Medv.med
#得21.2
```

In [ ]:

```

```