1. $\hat{f}(x_0) = x_0^T \cdot \hat{\beta}$

$E[\hat{f}(x_0)] = E[x_0^T (X^T X)^{-1} X^T \vec{y}]$ 注意到 $X = \begin{bmatrix} x_0^T \\ \vdots \\ x_n^T \end{bmatrix}$

$y(x_0) = x_0^T \beta^*$

$\vec{y} = \begin{bmatrix} y(x_0) \\ y(x_1) \\ \\ y(x_n) \end{bmatrix} = \begin{bmatrix} x_0^T \beta^* \\ x_1^T \beta^* \\ \\ x_n^T \beta^* \end{bmatrix} = X\beta^*$

$E[\hat{f}(x_0)] = E[x_0^T (X^T X)^{-1} X^T X \beta^*] = x_0^T \beta^* = y(x_0)$

即 $bias(\hat{f}(x_0)) = E[\hat{f}(x_0)] - y(x_0) = 0$

x_i $p \times 1$ X: $n \times p$ y: $n \times 1$ $X^T X$: $p \times p$

(2) 法二 $\hat{f}(x_i) = x_i^T \hat{\beta} = x_i^T \cdot (X^T X)^{-1} X^T y$

X是自变量，不具有随机性，只有y有随机性

$Var(\hat{f}(x_i)) = x_i^T (X^T X)^{-1} X^T \, Var(y) \, X [(X^T X)^{-1}]^T x_i$

$1\times p \quad p\times p \quad p\times n \quad n\times n \quad n\times p \quad p\times p \quad p\times 1$

$Var(y) = \begin{pmatrix} \sigma^2 & & & \\ & \sigma^2 & & \\ & & \sigma^2 & \\ & & & \ddots \\ & & & & \sigma^2 \end{pmatrix}_{n} = diag(\sigma^2, \sigma^2 \cdots \sigma^2)$

$\therefore Var(\hat{f}(x_i)) = \sigma^2 \cdot [x_i^T (X^T X)^{-1} X^T X [(X^T X)^{-1}]^T x_i]$

$\sum_{i=1}^{n} Var(\hat{f}(x_i)) = \sigma^2 \{ \sum_{i=1}^{n} x_i^T [(X^T X)^{-1}]^T x_i \}$

$= \sigma^2 \{ \sum_{i=1}^{n} x_i^T (X^T X)^{-1} x_i \}$

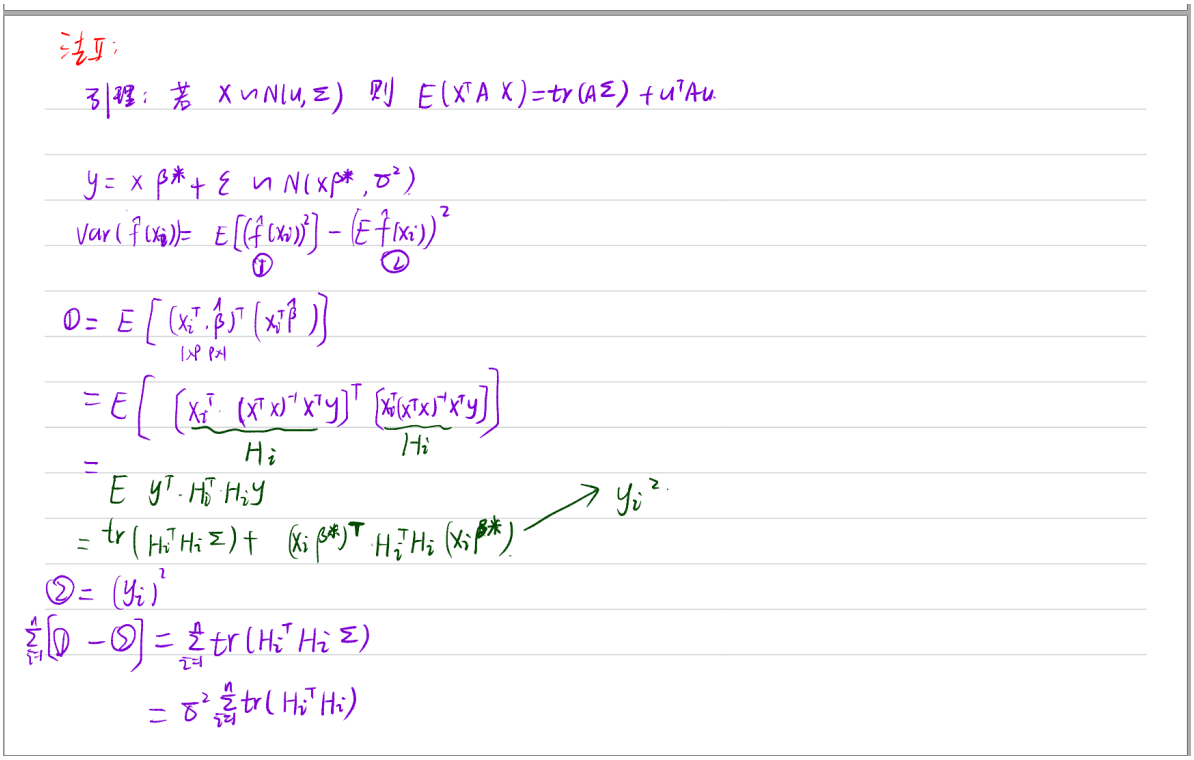
$(X^T X)^{-1}$ 对称 正定. 令 $(X^T X)^{-1} = A = Q^T \Lambda Q = P^T P$ $P = (X^T X)^{-\frac{1}{2}}$ $P x_i = (X^T X)^{-\frac{1}{2}} x_i$

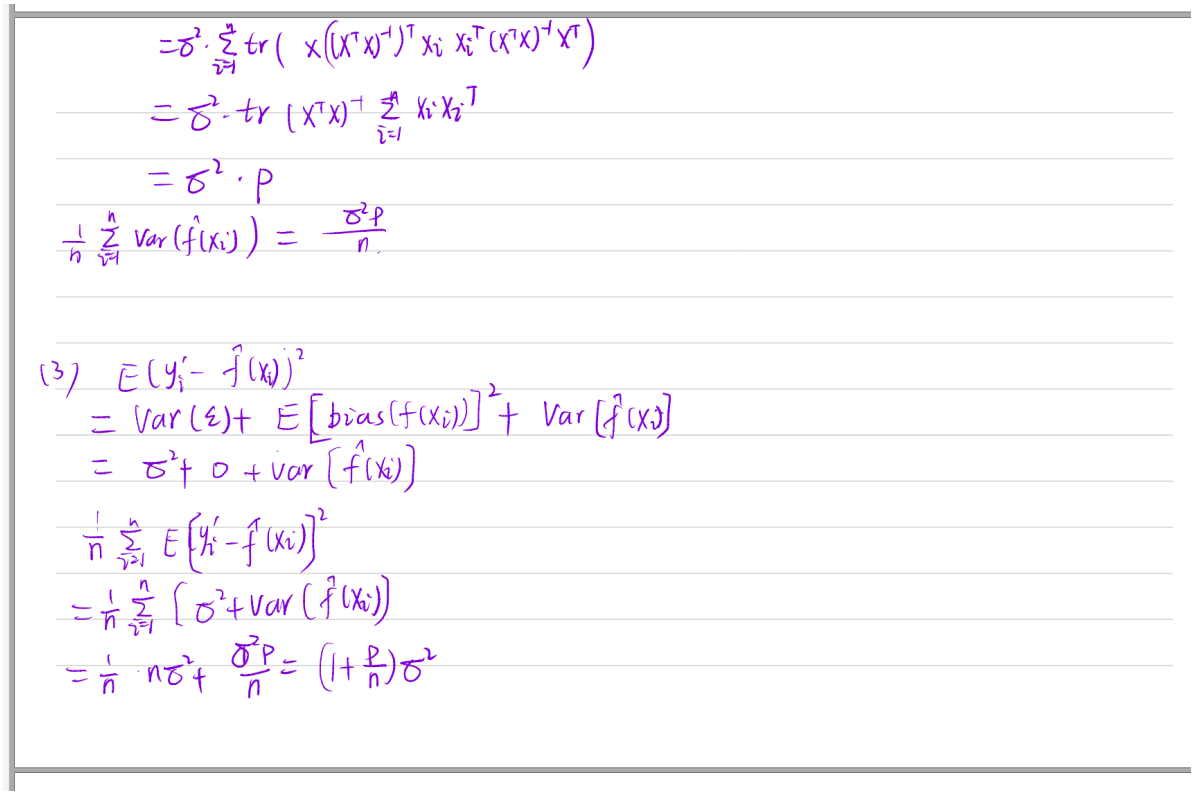
$\sum_{i=1}^{n} Var(\hat{f}(x_i)) = \sigma^2 [\sum_{i=1}^{n} x_i^T P^T P x_i]$

$= \sigma^2 \, tr \sum_{i=1}^{n} x_i^T P^T P x_i$

$= \sigma^2 \cdot tr(X^T P^T P X) = \sigma^2 p$

$\frac{1}{n} \sum_{i=1}^{n} Var(\hat{f}(x_i)) = \frac{p}{n} \sigma^2$

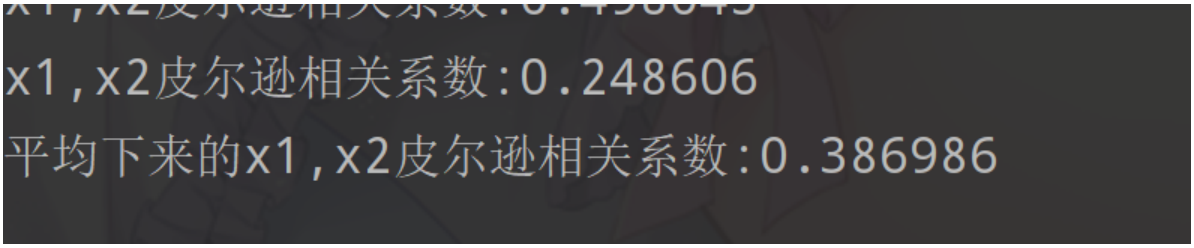




10_2

(1) 请由仿真结果计算得到$x1$与$x2$的皮尔森相关系数

10000次仿真结果的平均值是：



x_1和x_2的皮尔逊相关系数是0.386986

(2) 请多次生成数据，观察正则化系数为 1 情况下三种模型拟合参数的稳定性

生成30次数据。

线性回归

线性回归的斜率和截距的均值及方差

```
beta1的平均值:3.016474,beta0的平均值:0.054433
beta1的方差:0.009815,beta1的方差:0.835609
```

线性回归的斜率的均值是3.016 ,方差是0.0098

截距的均值是0.0544，方差是0.835

岭回归

岭回归的斜率和截距的均值及方差

```
beta1的平均值:2.990340,beta0的平均值:0.196614
beta1的方差:0.013563,beta1的方差:0.653476
```

岭回归的斜率的均值是2.990 ,方差是0.0136

截距的均值是0.197，方差是0.653

lasso回归

lasso回归的斜率和截距的均值及方差

```
beta1的平均值:2.939963,beta0的平均值:0.003490
beta1的方差:0.009823,beta1的方差:0.000353
```

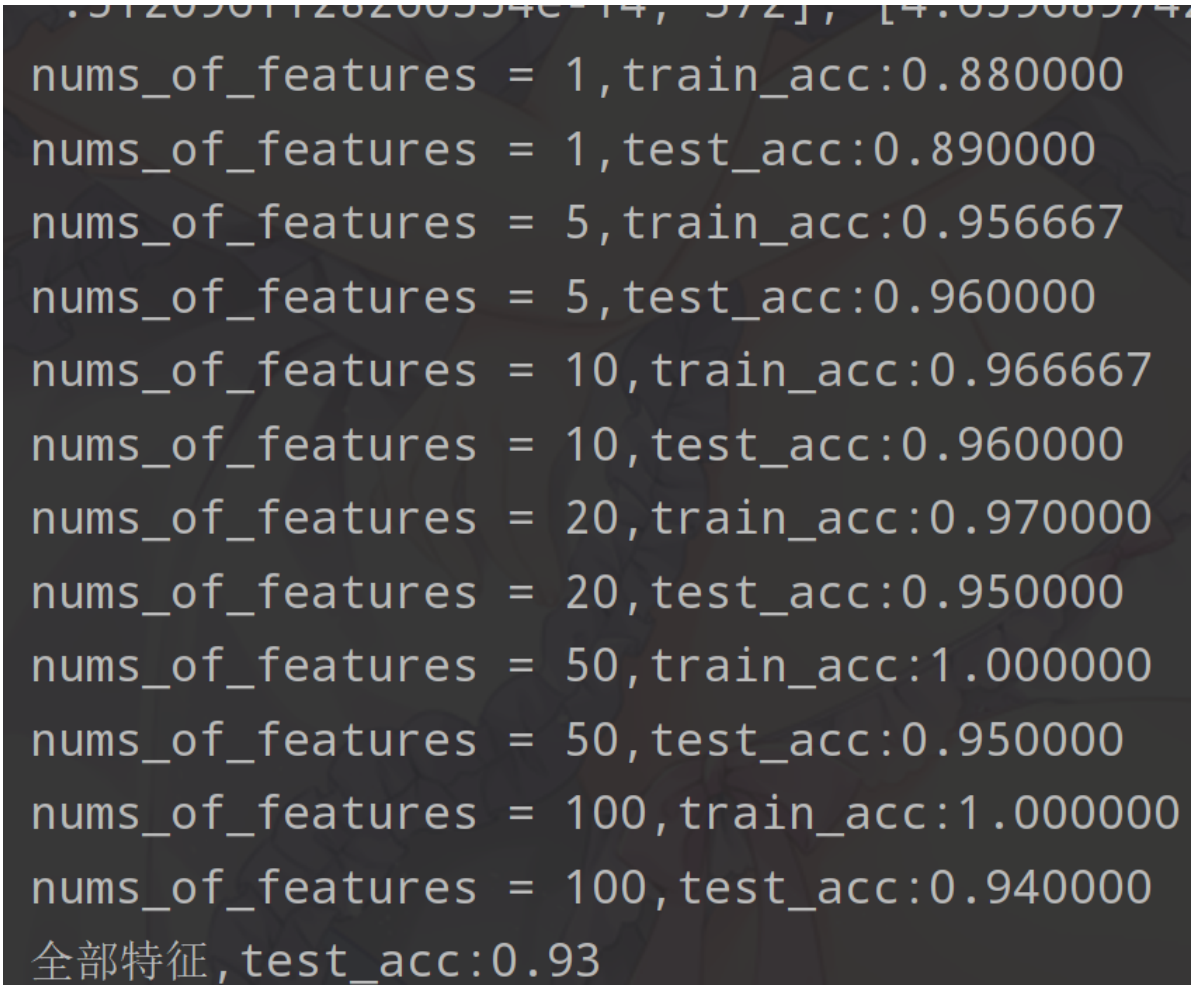
lasso回归的斜率的均值是2.940 方差是0.0098

截距的均值是0.0035 方差是0.000353

不难发现，lasso的截距和斜率的方差是最小的，说明lasso对于此题的稳定性最强，岭回归的斜率的稳定性不如线性回归，但是截距的稳定性高于线性回归。

10_3

采用fisher法



特征数	训练集上的acc	测试集上的acc
1	0.88	0.89
5	0.957	0.96
10	0.967	0.96
20	0.97	0.95
50	1	0.95
100	1	0.94
全部特征	1	0.93

显然，做特征选择后预测结果有所上升。原因主要是通过特征选择减少了过拟合，提高了模型的泛化能力，减少了不必要的特征对于分类的影响。

最大信息系数

```
nums_of_features = 1,train_acc:0.880000
nums_of_features = 1,test_acc:0.890000
nums_of_features = 5,train_acc:0.963333
nums_of_features = 5,test_acc:0.960000
nums_of_features = 10,train_acc:0.973333
nums_of_features = 10,test_acc:0.950000
nums_of_features = 20,train_acc:0.970000
nums_of_features = 20,test_acc:0.960000
nums_of_features = 50,train_acc:1.000000
nums_of_features = 50,test_acc:0.880000
nums_of_features = 100,train_acc:1.000000
nums_of_features = 100,test_acc:0.910000
全部特征,test_acc:0.93
```

类似的表：

特征数	训练集上的acc	测试集上的acc
1	0.88	0.89
5	0.963	0.96
10	0.973	0.95
20	0.97	0.96
50	1	0.88
100	1	0.91
全部特征	1	0.93

与fisher法的类似，都有过拟合的样子。

除此之外，请比较两种方法在这些特征个数时挑选出的特征子集有多少特征是相同的：

先找出选择多少特征的时候效果比较好：可以看出5个特征或者10个特征的时候效果比较好。

在这个特征数下，寻找相同的特征：

	5	10
fisher	[47, 916, 4, 219, 415]	[47, 916, 219, 4, 415, 476, 271, 224, 825, 461]
最大化信息	[47, 916, 4, 219, 415]	[47, 916, 219, 4, 415, 835, 468, 407, 747, 634]

两者在5个特征的时候，选择的特征一模一样，在10个特征的时候，选择的特征只有前5个一样，后面的都不一样。

这说明在选择特征较少时，可能选择的特征是一样的，但是当选择的特征较多时，选择的特征可能会出现差异。

前向算法

1）数据有400个，每一个的都有1000个特征

2）初始化一个空列表M_0，M_i表示存放了i个特征。

3）已知M_k的时候，剩下1000-k个特征，这时候把每个特征分别加入M_k中，用logisitic回归子啊训练集上训练一次，然后在验证集上看看效果，计算RSS等参数

4）选择RSS最好的特征，与之前的M_k一起，变成M_{k+1}

5）在$M_1, M_2, M_3, , , M_{1000}$中，选择RSS最高的提取出来。

实验结果

当特征选到6个的时候，R^2已经变成了1,继续是实验下去也没有意义了，所以可以直接结束了

特征数	R^2	选择的特征
1	0.196	47
2	0.464	47，4
3	0.732	47，4，103
4	0.866	47，4，103，283
5	0.866	47，4，103，283，0
6	1	47，4，103，283，0，134

与上一题的对比，发现差别还是不小的，这前5个选择之中，只有47和4是两者共有的，这说明算法之间还是存在差异的。我认为这里的差异主要是异步算法的问题。无论是fisher还是最大信息系数，都是在同步计算特征的影响，但前向算法是在固定了前面的特征的情况下，观察后续特征的。这可能会造成比较大的选择差异。

决策树算法

决策树算法会自动选择特征，用训练集的数据训练决策树，最后选择的特征是

```
[[47, 0.6331313267176789], [4, 0.1891946068699743], [552, 0.04640269719829376],
[916, 0.02853904680203866], [311, 0.02690100430416069], [851,
0.017595625456809486], [117, 0.01743583612306709], [217, 0.012777977044476323],
[89, 0.010087876614060258], [856, 0.008967001434720229], [943,
0.008967001434720229]]
```

与（1）相比，共同选取了47，4，916这三个特征。