**Boston房价回归课程报告**

学号：20307110315

姓名：周训哲

# 1.任务描述

主要采用ridge regression和lasso regression的理论模型自己实现了ridge和lasso库中的最主要的两个函数：fit（）和predict（）。除此以外，还尝试调用sklearn的库，利用已有模型实现了利用linear回归，ridge回归，lasso回归，梯度提升法回归，SVR中的支持向量机（SVM）的线性核函数回归，多项式核回归，决策树回归的方法，完成了对Boston多因子房价的预测。最后通过设置相同训练集和测试集（8：2），控制变量比较了不同方法的测试结果的准确率（以R^2和均方根误差作为判别标准），发现梯度提升和决策树方法效果最好。通过交叉验证法，发现对于ridge回归，alpha取0.012时的拟合效果最好，对于lasso回归，alpha取0.0011时的拟合效果最好。

# 2.数据集描述

**采用何种数据集开展实验？**

采用波士顿房价数据集。波士顿房价数据集来源于1978年美国某经济学杂志上。该数据集包含若干波士顿房屋的价格及其各项数据，每个数据项包含14个数据，分别是犯罪率、是否在河边和平均房间数等相关信息，其中最后一个数据是房屋中间价。

**数据集有什么特点？**

该数据集为多因子统计数据，总共包含14种房子相关的因素以及评分：

1.CRIM: per capita crime rate by town 每个城镇人均犯罪率

2.ZN: proportion of residential land zoned for lots over 25,000 sq.ft. 超过25000平方英尺用地划为居住用地的百分比

3.INDUS: proportion of non-retail business acres per town 非零售商用地百分比

4.CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise) 是否靠近查尔斯河

5.NOX: nitric oxides concentration (parts per 10 million) 氮氧化物浓度

6.RM: average number of rooms per dwelling 住宅平均房间数目

7.AGE: proportion of owner-occupied units built prior to 1940 1940年前建成自用单位比例

8.DIS: weighted distances to five Boston employment centres 到5个波士顿就业服务中心的加权距离

9.RAD: index of accessibility to radial highways 无障碍径向高速公路指数

10.TAX: full-value property-tax rate per $10,000 每万元物业税率

11.PTRATIO: pupil-teacher ratio by town 小学师生比例

12.B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town 黑人比例指数

13.LSTAT: % lower status of the population 下层经济阶层比例

14.MEDV: Median value of owner-occupied homes in $1000's 业主自住房屋中值

**数据集的各项统计量是多少？**

总共包含了506组房价统计的信息，其中每组信息包含14项数据。

# 3.方法介绍

## 3.1数据预处理（如有）

**从原始数据到输入模型的数据之间，经过了哪些处理？这些处理有何作用？**

对于数据的获取有两种方式，一种是下载为本地csv，通过read\_csv（）函数读取本地文件，并且将数据进行列的命名。另一种是用load\_boston（）函数从网上读取数据，用.data和.target得到自变量和因变量。

在实验过程中，由于没有单独分开的训练集和测试集，人为随机将数据集利用train\_test\_split（）函数拆分为80%的训练集和20%的测试集，根据训练集拟合的函数对测试集数据进行回测，再用预测数据与测试数据进行比对和计算损失函数，得到拟合结果的优劣。

## 3.2算法描述

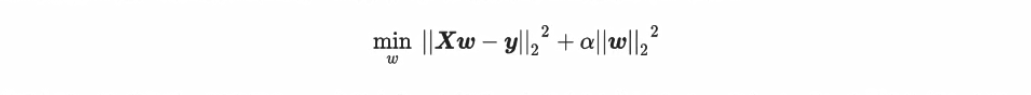
**算法原理**：

Linear Regression原理：

利用最小二乘法，使得系数矩阵与相关因子相乘得到的预测值与实际值相差的损失函数最小。具体方法就是首先计算预测值的大小Y^=ω1​X1​+ω2​X2​+…+ω13​X13​+b，然后引入损失函数，其中最为常用的是均方误差（MSE）：MSE=1/n\*​i=1∑n​(Yi​^​−Yi​)^2.

对损失函数进行优化的方法一般采用梯度下降法，即对损失函数求导，得到损失函数在某一区间的极小值。xn​+1=xn​−λ▽f(x),n≧0。最后根据梯度下降法求得的系数，计算预测值，与真实值进行比对，判断准确率。

Ridge Regression原理：

为了防止过拟合现象的发生，岭回归在多元线性回归之上增加了正则项，表达式为系数w的L2范式（即w的平方项）乘以正则化系数alpha。其损失函数写做

即 （系数为theta，自变量为A，正则化系数为lamda）

损失函数对系数求导即得

即。

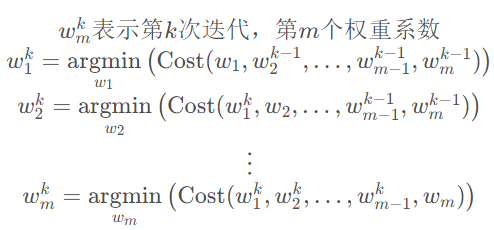
Lasso Regression原理：

和Ridge Regression一样，也加上了正则项，只是将正则项改成了带惩罚系数alpha关于系数w的L1范式（绝对值），由于是绝对值，所以不是处处可导，没办法通过直接求导得到w的解析解。我采用了较为简单的坐标下降法来求解系数w。

坐标下降法的原理即沿着一个坐标轴的方向，通过不断地迭代更新系数的值来逼近最优解。具体步骤：  
（1）初始化权重系数 w，例如初始化为零向量。

（2）遍历所有权重系数，依次将其中一个权重系数当作变量，其他权重系数固定为上一次计算的结果当作常量，求出当前条件下只有一个权重系数变量的情况下的最优解。

在第 k 次迭代时，更新权重系数的方法如下：



（3）步骤（2）为一次完整迭代，当所有权重系数的变化不大或者到达最大迭代次数时，结束迭代。

**整体流程：**

首先导入数据集，然后将数据集进行人为拆分为训练集和测试集，调用库函数，建立一个回归模型，调用模型中的fit（）函数对训练集数据进行拟合。然后用模型中的predict（）函数对测试集进行预测，得到预测值。

**函数描述及优化：**

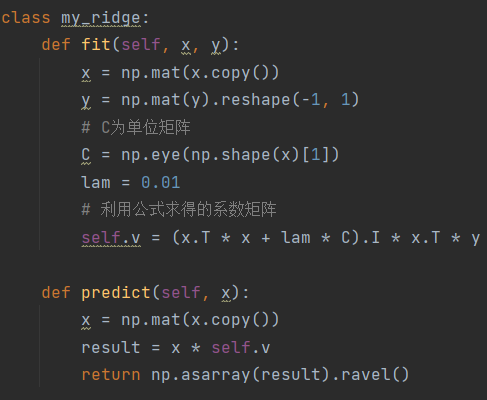
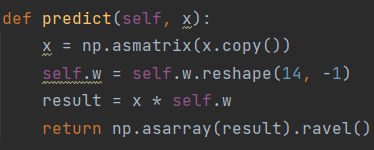
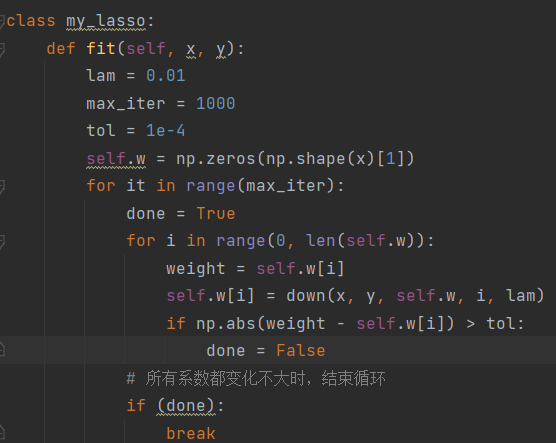
映射函数：对于变量data与价格price而言，则是一个多项式回归，f（x）=w1\*x1^t1+…wn\*xn^tn…。

损失函数：为了拟合的程度尽量的高，则需要寻求合适的系数w。引入损失函数f（w），最常用的多项式线性回归的损失函数，则是均方误差法，即将y^=wx与实际的y相减的差值平方并加和求平均得到损失函数，这个也是最简单的线性回归的损失函数。对于岭回归，其损失函数引入一个含lamda项的关于w的L2正则表达式；对于lasso回归则引入一个含lamda项关于w的L1正则表达式。具体对于岭回归和lasso回归的损失函数的优化，则在于正则化系数lamda的选择。可以通过交叉验证法，对不同的lamda进行比较拟合求r^2来判断拟合程度，来寻求一个最好的正则化系数。

**训练及预测过程：**

训练过程即利用训练集进行拟合（fit（）），寻求损失函数最小的系数值，即完成训练过程。预测过程即将训练过程得到的系数对测试样例进行映射（predict（）），得到预测值，将预测值与实际值进行比对，运用打分函数（求r^2以及均方根误差）来判断拟合程度的高低。

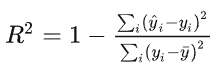
**算法描述：**

****

# 4.实验结果分析

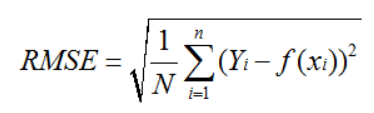
## 4.1评价指标

评价指标之一为拟合系数（R2）



用来判断拟合程度的高低，当R2越接近1，说明y^越接近y，即预测值越接近真实值，说明拟合程度很好。

评价指标之二为均方根误差



当均方根误差越大时，说明预测值与实际值的误差越大，拟合程度越低。

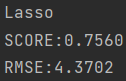
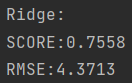
## 4.2定量评价结果



同时设定测试集与训练集分配的比例与随机分配的随机种子相同来进行测试，以控制数据集相同，以防非算法本身的误差所导致结果不同。

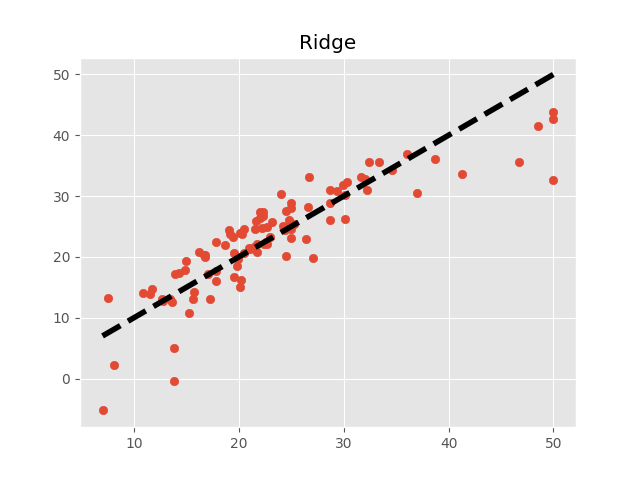
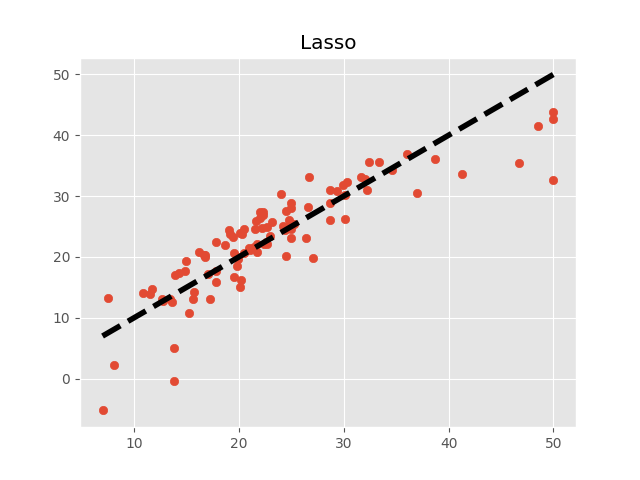


计算r2以及均方根误差的结果并输出：

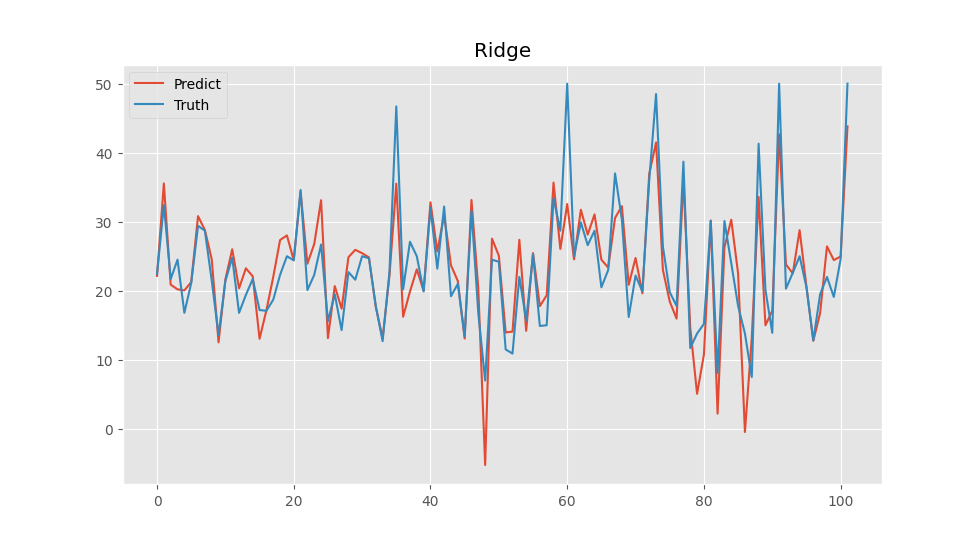
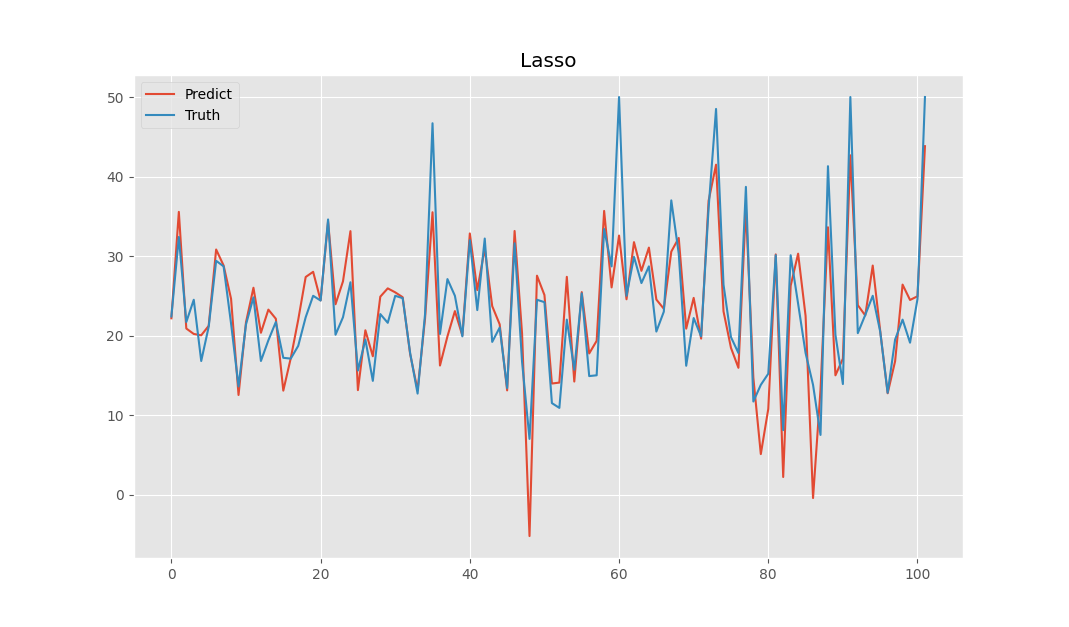


这个是在正则化指数都为0.01的条件下得出的结果，可以看出两者差别并不是很大，拟合程度都较高。

## 4.3可视化结果

该图片横坐标为房价实际值，纵坐标为房价预测值，黑色虚线为y=x，当点阵集中在黑色虚线附近时说明预测值与真实值相接近，拟合程度较好。

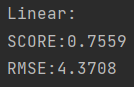
 

该图片将预测值与实际值根据顺序作为纵坐标放在一张图里，当两条曲线的拐点相差不大时说明预测准确。

由以上两种回归方法对比，可见对于每个数据的拟合差别也不是很大。

## 4.4补充及优化

由上文所述，线性回归最为简单，只需利用最小二乘法拟合即可，故我也自己根据公式实现了线性回归的函数，其表现如下：

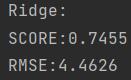
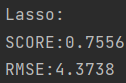
可见，其与ridge和lasso的回归差别也不是很大。

究其原因，其本质都是对线性回归的拓展，只不过增加了惩罚函数，当正则化系数等于零，则意味着与线性回归函数等同。由于在实验过程中lamda都取0.01偏小，故结果差别不大。

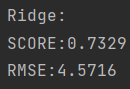
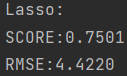
**正则系数**

下面将探索正则系数与拟合程度以及拟合系数得关系：



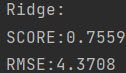
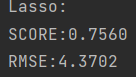
 



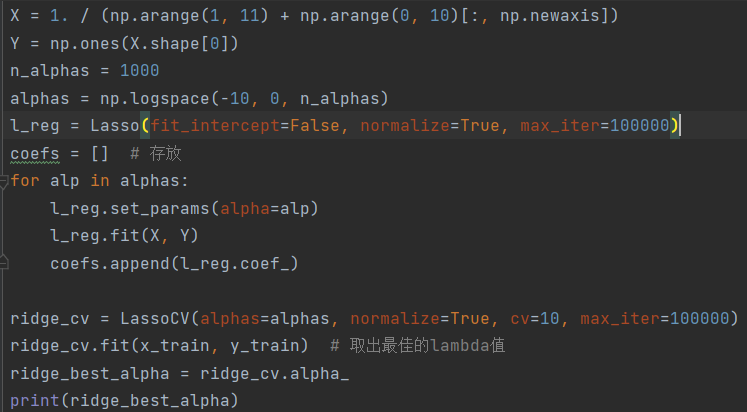
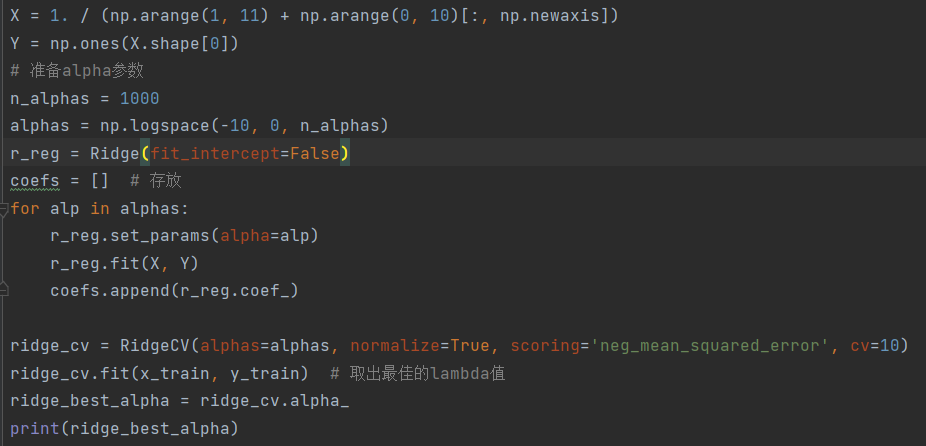
与之前对比可见，随lamda增大，其拟合程度在不断下降



随着lamda趋近0，其拟合结果越来越趋近于线性回归。

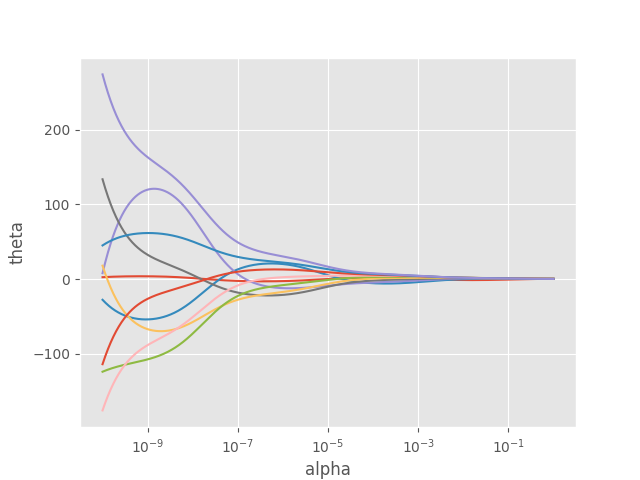
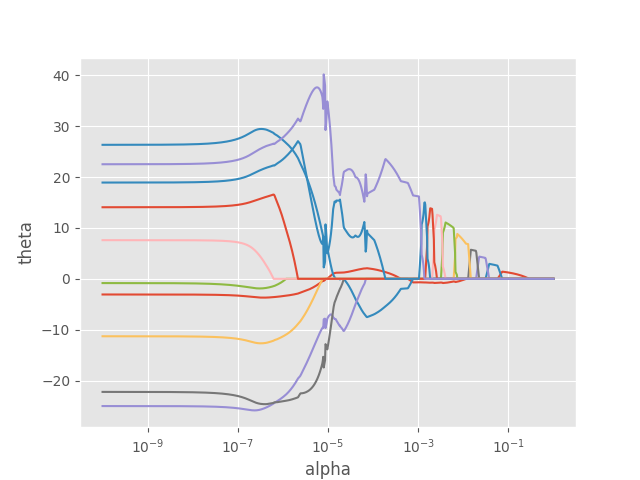
对于最佳lamda的取值，可以利用交叉验证的方法来对不同lamda进行遍历计算均方根误差，并进行排序，来寻找最优正则化系数。



最终得到ridge的最佳正则化系数值为

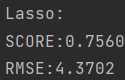
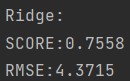
lasso的最佳正则化系数为

用可视化表示正则化系数与拟合系数的关系：

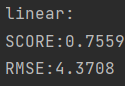
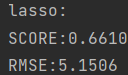
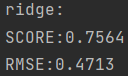
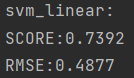
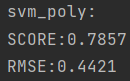
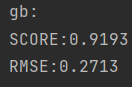
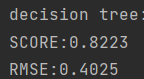
由图可见，当alpha增大时，拟合系数会收敛到0。则最佳的alpha即快收敛前的值。

将交叉验证得到的值带入之后

差别也不是很明显，分析原因，可能是lasso交叉验证时设置了排序score因子，而ridge没有使用，所以得到的结果并不是相对应需要的最优值。

**其他回归方法（导入sklearn库）：**

可见，利用梯度上升和决策树方法拟合的结果最好，其r2最高，均方根误差最小。

# 总结

通过对比以及学习，线性回归的基本原理即是最小二乘法，对于线性的变量十分有效。即使最小二乘估计是无偏的，它们的方差很大，但因多重共线性的存在，它们可能离真实值很远。

岭回归则加上一个惩罚项来最小化目标函数。alpha值越高，对系数的限制越多;alpha低则对系数几乎没有限制，其泛化能力更强，在这种情况下，线性回归和岭回归类似。欠拟合，则降低alpha值；过拟合，则增加alpha值。Lasso原理与岭回归类似，但是在岭回归中，再大的alpha是将系数无限逼接近于零但不等于零，这是Lasso不一样的地方。

而其他方法由于知识水平有限，暂时只知道如何从sklearn中调用，尚未能研究其回归的原理，故不作比较。

通过本次作业，我了解到对于数据回归的多种机器学习回归方法，除此以外还了解许多对数据的处理方法，学习了损失函数、梯度下降、正则化等机器学习的相关知识，为之后的机器学习回归方面的知识打下基础。

参考文献：

[1]<https://blog.csdn.net/weixin_45508265/article/details/113255859>

[2][https://blog.csdn.net/weixin\_30769833/article/details/114387761?ops\_request\_misc=&request\_id=&biz\_id=102&utm\_term=lasso%E5%9B%9E%E5%BD%92%E4%BA%A4%E5%8F%89%E9%AA%8C%E8%AF%81&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~sobaiduweb~default-0-114387761.142^v51^control,201^v3^add\_ask&spm=1018.2226.3001.4187](https://blog.csdn.net/weixin_30769833/article/details/114387761?ops_request_misc=&request_id=&biz_id=102&utm_term=lasso%E5%9B%9E%E5%BD%92%E4%BA%A4%E5%8F%89%E9%AA%8C%E8%AF%81&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-114387761.142%5ev51%5econtrol,201%5ev3%5eadd_ask&spm=1018.2226.3001.4187)

[3]<https://baike.baidu.com/item/Lasso%E7%AE%97%E6%B3%95/22685468>

[4]<https://baike.baidu.com/item/%E5%B2%AD%E5%9B%9E%E5%BD%92?fromModule=lemma_search-box>