

**机器学习报告**

**题目： 回归**

二级学院 计算机科学与工程学院

专 业 计算机科学与技术

班 级 119030703

学生姓名 王晓娟 学号 11912030134

指导教师 曹琼

时 间 2021.1.30

目 录

[目 录 I](#_Toc63888240)

[摘 要 II](#_Toc63888241)

[1、回归算法基本原理 3](#_Toc63888242)

[1.1 回归的基本概念 3](#_Toc63888243)

[1.2 回归的类型 3](#_Toc63888244)

[1.2.1 线性回归 3](#_Toc63888245)

[1.2.2 Ridge回归 3](#_Toc63888246)

[1.2.3 LASSO回归 4](#_Toc63888247)

[1.2.3 梯度下降算法 4](#_Toc63888248)

[1.2.4 Logistic回归 5](#_Toc63888249)

[1.2.5 Softmax回归 6](#_Toc63888250)

[2、算法和实现 8](#_Toc63888251)

[2.1 算法的设计步骤 8](#_Toc63888252)

[2.2 算法的基础模块 10](#_Toc63888253)

[2.2.1 梯度下降算法 10](#_Toc63888254)

[2.2.2 Logistic回归 11](#_Toc63888255)

[2.2.3 softmax回归 11](#_Toc63888256)

[2.3 算法与Sklearn调包实现对比 11](#_Toc63888257)

[2.3.1 线性回归使用直接公式法（使用波士顿房价数据集） 11](#_Toc63888258)

[2.3.2 Ridge回归使用直接公式法（使用波士顿房价数据集） 12](#_Toc63888259)

[2.3.3批量梯度下降算法（L2正则项）（使用波士顿房价数据集） 13](#_Toc63888260)

[2.3.4 随机梯度下降算法（L2正则项）（使用波斯顿房价数据集） 13](#_Toc63888261)

[2.3.4 mini-batch（L2正则项）（使用波斯顿房价数据集） 14](#_Toc63888262)

[2.3.5 lasso回归（L1正则项）（使用波斯顿房价数据集） 15](#_Toc63888263)

[2.3.5 logistic回归（使用乳腺癌数据集） 15](#_Toc63888264)

[2.3.5 Softmax回归（使用鸢尾花数据集） 15](#_Toc63888265)

[3、总结 15](#_Toc63888266)

[3.1 遇到的问题 15](#_Toc63888267)

[4.2 心得体会 16](#_Toc63888268)

摘 要

回归模型是机器学习中一种最基本的模型，其功能是建模和分析变量之间的关系。线性回归就是用线性函数来解决回归问题。本文对线性回归中的几种模型算法进行了续写。

关键词（3～5个）：线性回归、Logistic回归、Softmax回归

# 1、回归算法基本原理

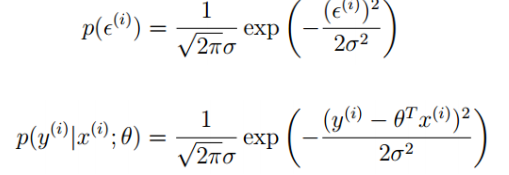
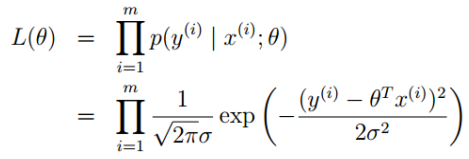
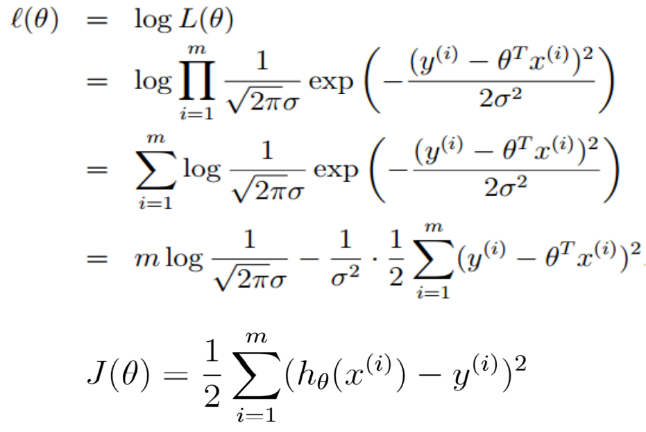
# 1.1 回归的基本概念

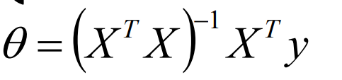
回归是从一组数据出发，确定某些变量之间的定量关系式，即建立数学模型并估计未知参数。通俗来讲就是给定一个点集，用一个函数去拟合这个点集，并且使得点集与拟合函数间的误差最小。

# 1.2 回归的类型

#### **1.2.1 线性回归**

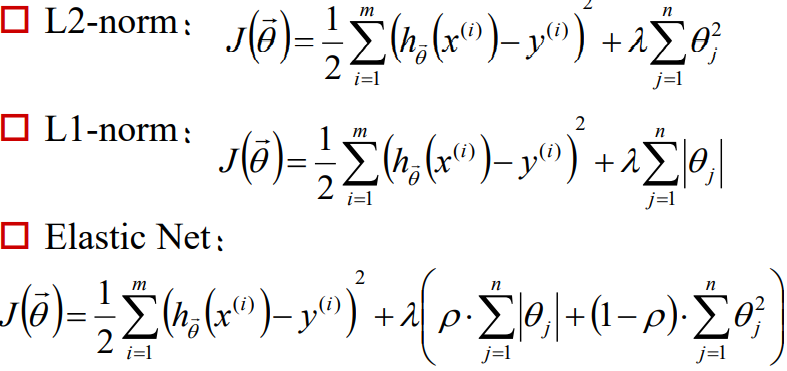
θTx（i）+ε（i），根据中心极限定理，误差是独立并且具有相同分布的，服从均值为0，方差为某定值的高斯分布。写出服从高斯分布的误差的概率密度函数

，现在相当于是已知观测所得的结果，对有关事物的性质的参数进行估计，这种形式就是似然函数。对似然函数求其最大值，先将其对数化。，

得到最小二乘法的损失函数，对J(θ)求偏导，得到参数的解析式

#### **1.2.2 Ridge回归**

将目标函数增加L1正则得到的模型叫做Lasso回归，增加L2正则叫做Ridge回归



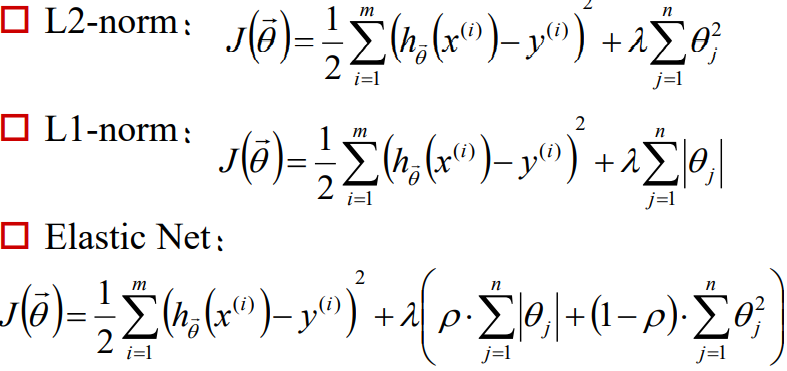
岭回归在最小二乘法的基础上加上了一个L2的惩罚项：

正规方程：*θ*=(*X*′*X*+*αI*)−1*X*′*Y*

Ridge回归在不抛弃任何一个变量的情况下，缩小了回归系数，使得模型相对而言比较的稳定，但这会使得模型的变量特别多，模型解释性差。

#### **1.2.3 LASSO回归**

Lasso回归有时也叫做线性回归的L1正则化，和Ridge回归的主要区别就是在正则化项，Ridge回归用的是L2正则化，而Lasso回归用的是L1正则化。Lasso回归的损失函数表达式如下：



Lasso回归使得一些系数变小，甚至还是一些绝对值较小的系数直接变为0，因此特别适用于参数数目缩减与参数的选择，因而用来估计稀疏参数的线性模型。

但是Lasso回归有一个很大的问题，就是它的损失函数不是连续可导的，由于L1范数用的是绝对值之和，导致损失函数有不可导的点。也就是说，我们的最小二乘法无法求解它。

#### **1.2.3 梯度下降算法**

对于数据量大，维度多的数据，使用公式直接算θ，效率低，耗费大，所以提出梯度下降算法。在微积分中，对多元函数的参数求θ偏导数，把求得的各个参数的导数以向量的形式写出来就是梯度。梯度就是函数变化最快的地方。梯度下降是迭代法的一种，在求解机器学习算法的模型参数θ时，即无约束问题时，梯度下降是最常采用的方法之一。

Gradient Descent(GD)

基本策略可以理解为”在有限视距内寻找最快路径下山“，因此每走一步，参考当前位置最陡的方向(即梯度)进而迈出下一步。

标准的梯度下降主要有两个缺点：

（1）训练速度慢：在应用于大型数据集中，每输入一个样本都要更新一次参数，且每次迭代都要遍历所有的样本，会使得训练过程及其缓慢，需要花费很长时间才能得到收敛解。

（2）容易陷入局部最优解：由于是在有限视距内寻找下山的反向，当陷入平坦的洼地，会误以为到达了山地的最低点，从而不会继续往下走。所谓的局部最优解就是鞍点，落入鞍点，梯度为0，使得模型参数不在继续更新。

Batch Gradient Descent(BGD)

BGD相对于标准GD进行了改进，改进的地方通过它的名字应该也能看出来，也就是不再是想标准GD一样，对每个样本输入都进行参数更新，而是针对一个批量的数据输入进行参数更新。

Stochastic Gradient Descent（SGD）

随机梯度下降法，不像BGD每一次参数更新，需要计算整个数据样本集的梯度，而是每次参数更新时，仅仅选取一个样本计算其梯度。

SGD的优点：

虽然看起来SGD波动非常大，会走很多弯路，但是对梯度的要求很低（计算梯度快），而且对于引入噪声，大量的理论和实践工作证明，只要噪声不是特别大，SGD都能很好地收敛。

应用大型数据集时，训练速度很快。比如每次从百万数据样本中，取几百个数据点，算一个SGD梯度，更新一下模型参数。相比于标准梯度下降法的遍历全部样本，每输入一个样本更新一次参数，要快得多。

SGD的缺点：

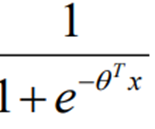
SGD在随机选择梯度的同时会引入噪声，使得权值更新的方向不一定正确（次要）。

SGD也没能单独克服局部最优解的问题（主要）

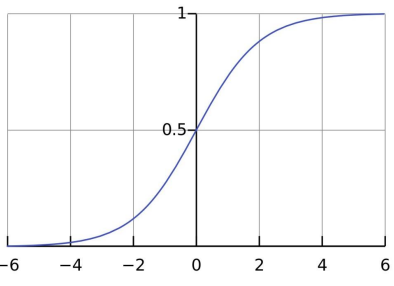
Mini-batch Gradient Descent（MBGD，也叫作SGD）

小批量梯度下降法就是结合BGD和SGD的折中，对于含有n个训练样本的数据集，每次参数更新，选择一个大小为 $m(m<n)$ 的mini-batch数据样本计算其梯度。

#### **1.2.4 Logistic回归**

逻辑回归，可以说是在线性回归的基础上加上一个sigmoid函数，将线性回归产生的值归一化到[0-1]区间内。线性回归与logistic回归的学习规则是相同的，不同之处在于线性回归中hθ （x） =θTX, 而logistic回归回归中hθ （x）=

P=hθ（x）=g(θTx)=表示y=1的概率，其中g（z）=



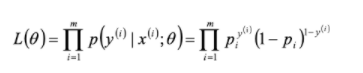
单个样本的概率分布：

P(y=1|x;θ)=hθ（x）

P(y=0|x;θ)=1-hθ（x）

P(y|x;θ)=(hθ（x）)y(1-hθ（x）)(1-y)

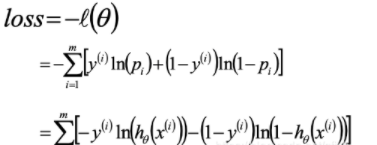
M个样本的概率分布——似然函数



对数似然函数

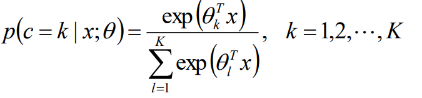


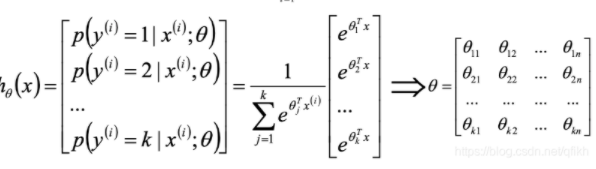
损失函数：



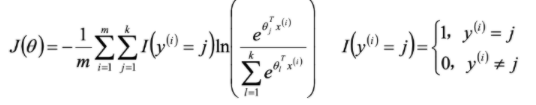
#### **1.2.5 Softmax回归**

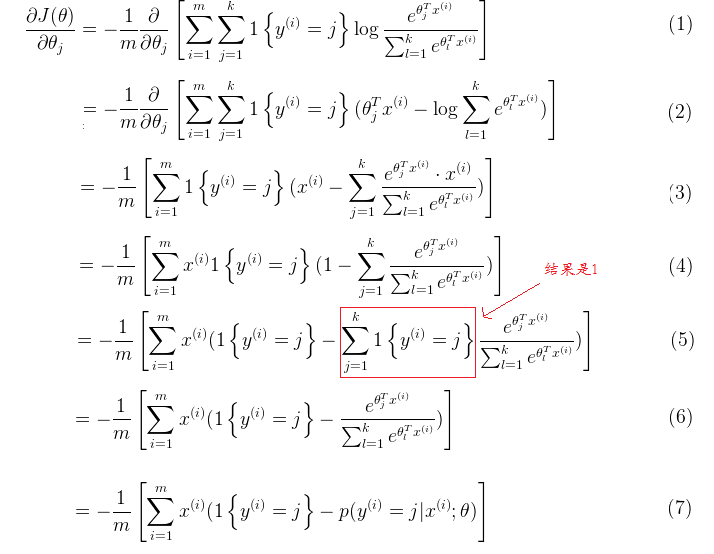
Softmax处理的是多分类问题

向量θ1将样本分为第一类和其他; 向量θ2将剩余的样本分为第二类和其他...θ是一个k\*n 的二维矩阵,将样本分成k类.单个样本的概率分布：



损失函数定义为：





对于 J ( θ ) \textstyle J(\theta)*J*(*θ*) 的最小化问题，目前还没有闭式解法。因此，我们使用迭代的优化算法（例如梯度下降法，或 L-BFGS）。经过求导，我们得到随机梯度下降公式如下：   


在梯度求导中，示性函数比较费事费力，一种比较简便的方法**是先将数据集转化为逻辑分类可以处理的数据结构。即为对象添加值为1的属性x0（截距项b），将输出分类y转换为one-hot编码。分类1表示为[1,0,0]，分类2表示为[0,1,0]，分类3表示为[0,0,1]，将处理后的y带入梯度公式。**

# 2、算法和实现

## 2.1 算法的设计步骤



图2.1.1 回归流程图

线性回归主要步骤：

1. 在数据后加一列1，方便计算wx+b。计算x的转置
2. 计算x的转置和x相乘的逆矩阵，再计算x的转置和y相乘的矩阵
3. 步骤2中的两个矩阵在进行相乘得到θ值。
4. 预测y的值，为x乘以θ在加上θ的最后一列。
5. 计算mse的值、RMSE的值、r2的值。

Ridge回归主要步骤：

1. 在数据后加一列1，方便计算wx+b。计算x的转置
2. 计算(（XTX）+α\*I)-1X-1Y
3. 预测y的值，为x乘以θ在加上θ的最后一列。
4. 计算mse的值、RMSE的值、r2的值。

梯度下降算法主要步骤：

1. 初始化参数
2. 增加截距项，在x后增加一列“1”
3. 计算损失
4. 计算梯度
5. 如果损失小于阈值，停止迭代，如果大于阈值，则θ-=eta0\*grad
6. 预测y的值，为x乘以θ在加上θ的最后一列。
7. 计算mse的值、RMSE的值、r2的值。

Logistic回归主要步骤：

1. 增加截距项，初始化θ
2. 计算梯度
3. 次数是否小于迭代次数，否则更新θ
4. 预测y的值，为x乘以θ在加上θ的最后一列。循环所有的预测值，判断是否大于0.5，如果大于0.5则为1，否则为0
5. 循环所有的预测值，判断准确率

Softmax回归：

1. 初始化参数
2. 增加截距项，初始化θ
3. 判断y有几个分类，按照分类树生成对角线为1的单位阵
4. 将y输出分类转化为one——hot编码
5. 将整理后的y带入梯度公式
6. 预测y的值，取每行最大的y值，然后返回最大值的位置索引就是分类类别

## 2.2 算法的基础模块

#### 2.2.1 梯度下降算法



图2.2.1.1 梯度下降算法流程图

梯度下降算法中每一次的θ更新为θ-=eta0\*grad，梯度等于xT\*（θx-y）/len(x)(线性回归梯度)

Ridge回归（梯度：（xT\*（θx-y）+α\*θ）/len(x)）

梯度下降结束的条件：

设置迭代次数、设置阈值

#### **2.2.2 Logistic回归**

Logistic回归使用的是梯度下降算法，不同之处在于线性回归中hθ （x） =θTX, 而logistic回归回归中hθ （x）=，在logistic回归中，要将求得的x\*θ映射到sigmoid函数中，logistic回归中也有正则化项。

Logistic回归中要得到的预测值映射到sigmoid函数中，然后在循环所有的预测值，判断得到的预测值与0.5的大小，以此做到二分类。

#### **2.2.3 softmax回归**

Softmax回归中使用one——hot编码形式，**先将数据集转化为逻辑分类可以处理的数据结构。即为对象添加值为1的属性x0（截距项b），将输出分类y转换为one-hot编码。分类1表示为[1,0,0]，分类2表示为[0,1,0]，分类3表示为[0,0,1]，将处理后的y带入，整理后的梯度下降公式为（XT\*(x\*θ-y\_onehot)）+α\*θ。**

**Softmax回归中要注意的是预测完y值后，返回的是**每行最大值的位置，位置索引就是分类。

## 2.3 算法与Sklearn调包实现对比

#### **2.3.1 线性回归使用直接公式法（使用波士顿房价数据集）**

使用线性回归的直接公式（使用相同的数据集）：

使用算法:

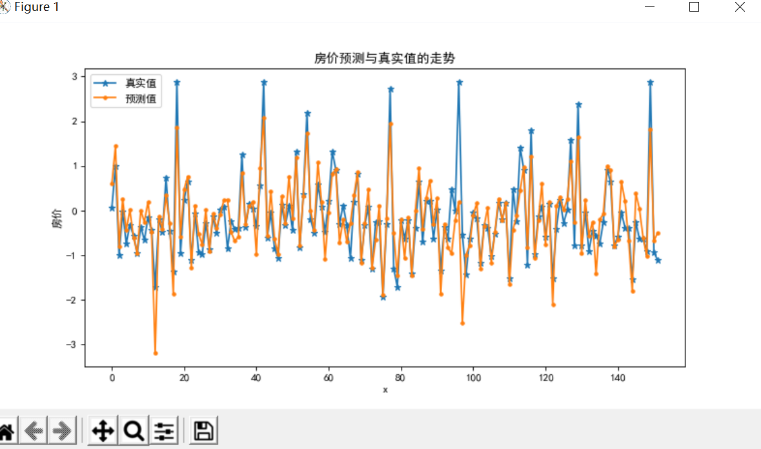


图2.3.1.1 线性回归房价预测与真实值的走向

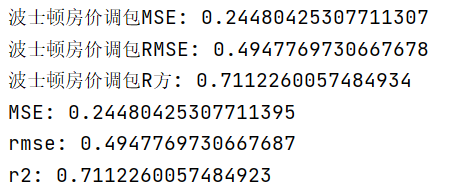


图2.3.1.2 线性回归房价算法对比

根据对比，得出sklearn调包和算法实现的一样

#### **2.3.2 Ridge回归使用直接公式法（使用波士顿房价数据集）**

使用Ridge直接公式法之后的决策树（使用相同的数据集）：

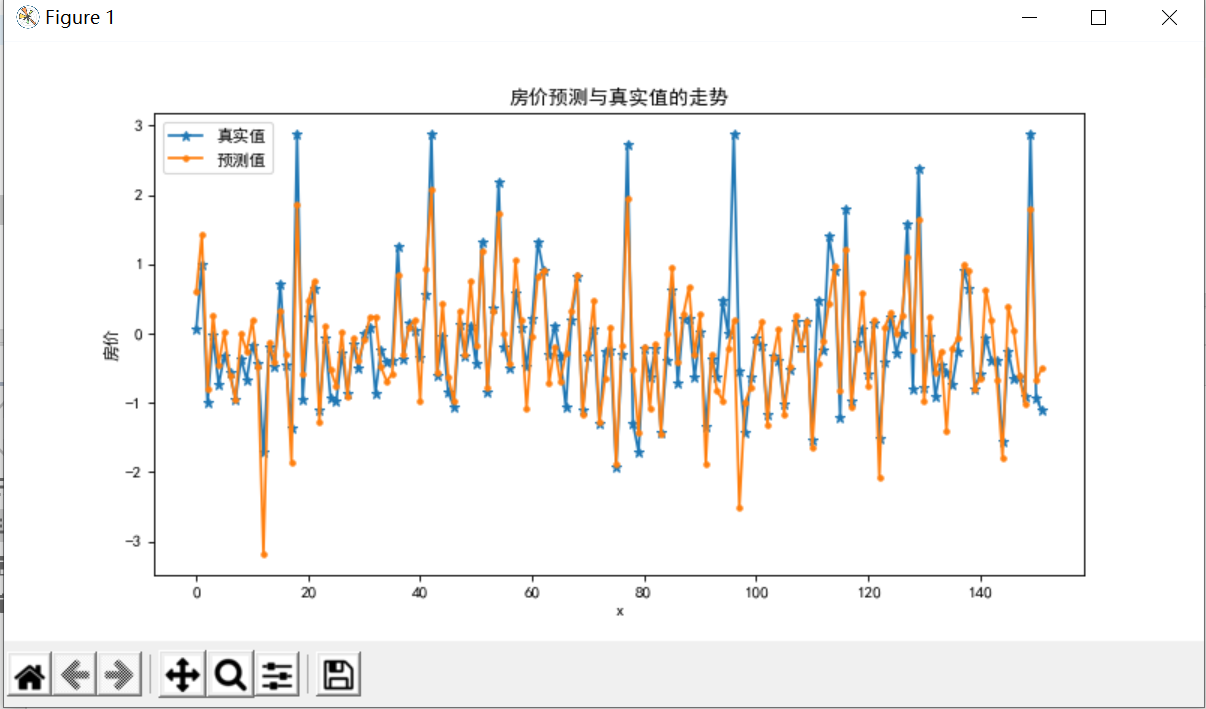


图2.3.2.1 Ridge回归房价预测与真实值的走向

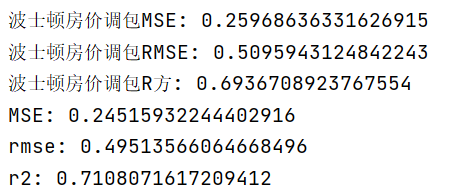


图2.3.2.2 Ridge回归房价算法对比

根据对比，得出sklearn调包和算法实现的一样

#### **2.3.3批量梯度下降算法（L2正则项）（使用波士顿房价数据集）**

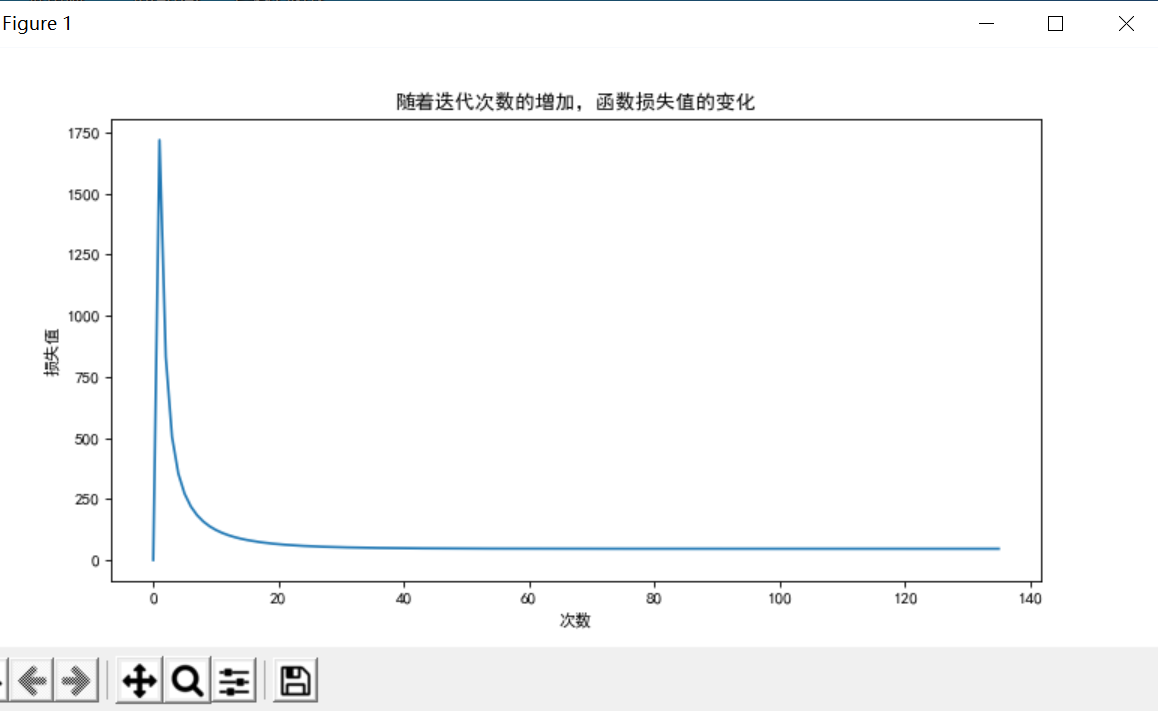


图2.3.3.1批量梯度下降算法

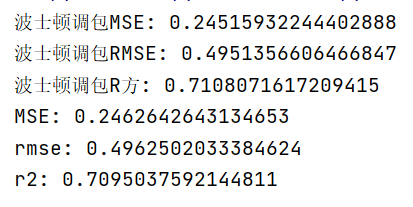


图2.3.3.1算法对比

#### **2.3.4 随机梯度下降算法（L2正则项）（使用波斯顿房价数据集）**

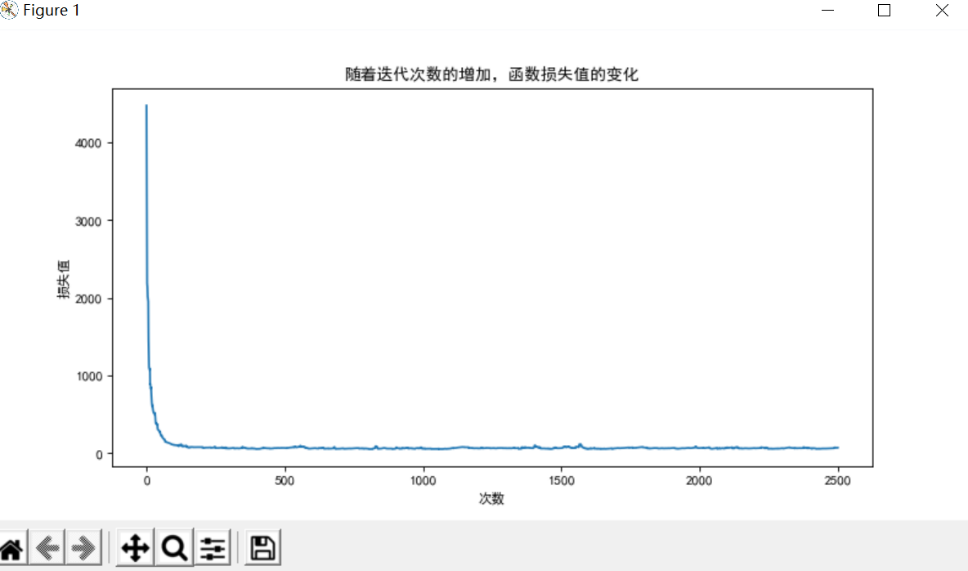


图2.3.4.1随机梯度下降算法

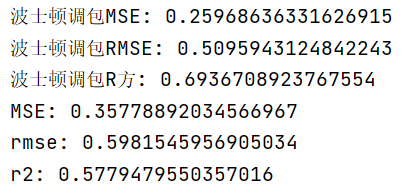


图2.3.4.1算法对比

#### **2.3.4 mini-batch（L2正则项）（使用波斯顿房价数据集）**

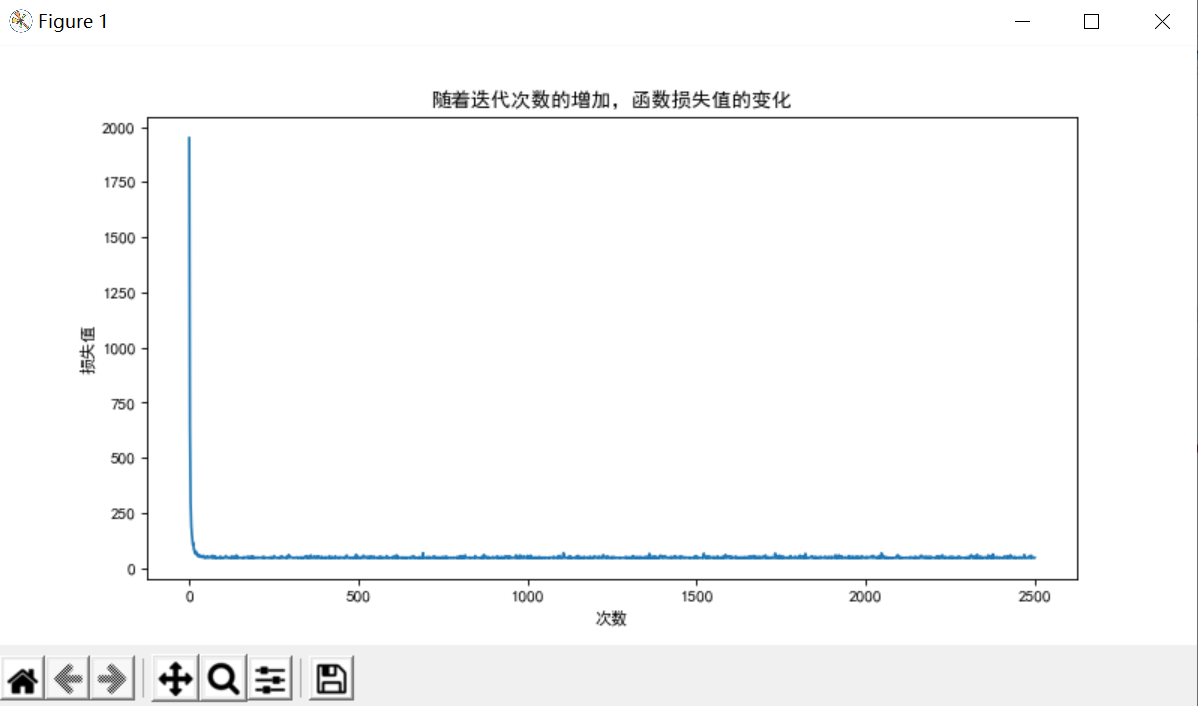


图2.3.4.1 mini-batch算法

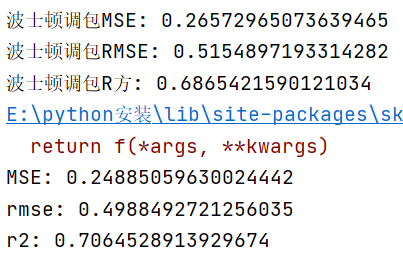


图2.3.4.1算法对比

#### **2.3.5 lasso回归（L1正则项）（使用波斯顿房价数据集）**

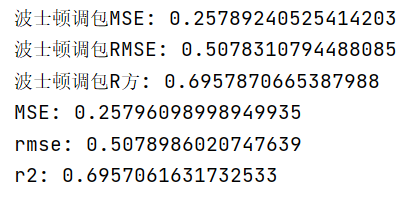
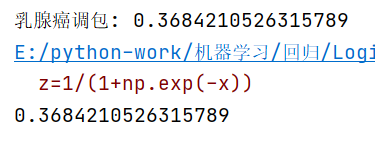
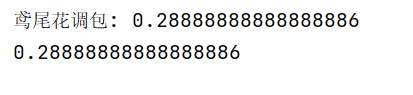


图2.3.5.1算法对比

#### **2.3.5 logistic回归（使用乳腺癌数据集）**



#### **2.3.5 Softmax回归（使用鸢尾花数据集）**



# 3、总结

## 3.1 遇到的问题

1.在写lasso回归的时候，lasso回归不同于线性回归和岭回归，就是它的损失函数不是连续可导的，由于L1范数用的是绝对值之和，导致损失函数有不可导的点，所以lasso回归不能使用最小二乘法。

2.在使用波士顿房价时，算法正确，但时mse始终过大，原因时数据没有进行归一化处理。

3.在梯度下降算法中，如果只设置迭代结束的条件是迭代次数，往往会有所浪费，如果损失达到一定阈值，迭代也可以停止。

4.在softmax回归中，得到的y\_hat不能直接作为预测值，而应该是选择每一行中最大值的索引作为分类值。

## 4.2 心得体会

通过这次的算法实现，学到了很多，对决策树有了更深一步的理解