机器学习第二次作业：正则化研究

统计81 于越 *

March 28, 2021

Abstract

这次作业主要涉及对于机器学习回归拟合中正则项的研究。当对于一些训练数据集进行拟合时，如果采用多项式拟合的形式，若多项式首项次数选择得过大，很容易会出现过拟合的情况。出现这一问题的根源在于拟合时的代价函数是实际值与拟合值的最小二乘形式，为了使得代价函数达到最小，形成的数据曲线会对应尽可能通过每一个点，导致各项权重不够合理。显然，过拟合的训练结果是不适合进行后续数据的预测的。因此，我们可以通过引入正则项的方法来解决过拟合问题。这次作业主要涉及了对于L2范数正则与L1范数正则的研究，并且对于带噪声的数据集进行不加正则项、加L2正则项与加L1正则项三次不同的拟合，比较拟合结果，并且选用了一系列的验证数据集用于验证实验结果。最后计算随着验证次数增加的验证过程中形成的误差平凡和进行比较，相应地得出不同正则项结果的比较。

1 关于L2正则项

1.1 相关公式

L2范数正则项，也就是在原来最小二乘代价函数的末尾增加价值向量的2范数，从而在实际过程中对于过拟合状况下的价值向量权重起到一定的“压缩”的效果，从而达到抵消过拟合效果的目的。原表现度量:

$$\sum_{i=1}^{n}(f_w(x_i)-y_i)^2 \tag{1}$$

原价值向量表达式:

$$w=(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}Y \tag{2}$$

值得一提的是，在加上L2范数正则项后，通过对价值向量w求偏导，仍旧可以直接得到对应的解析解的形式: 加L2正则项后表现度量:

$$\sum_{i=1}^{n}(f_w(x_i)-y_i)^2+\lambda*||w||_2 \tag{3}$$

*学号：2183210516

加L2正则项后价值向量表达式:

$$w = (X^{\mathrm{T}}X + \lambda * I)^{-1}X^{\mathrm{T}}Y \tag{4}$$

注意上式中的单位阵I的第一行第一列元素要变为0，这是因为我们默认对于常数项不需要正则化处理。

1.2 实验结果呈现及分析

在这里，我们使用了分布大致为

$$(x^4 - 1000x^3 + 5x^2 - 200x + 150)/2 * 10^6 \tag{5}$$

的训练数据集，并对这些数据集加以正态分布噪声。在这里，我们选择了300组训练集数据进行训练，并且选择了100组验证集数据进行验证。最后我们做出了对于验证集的随着验证次数增加的拟合误差的图像.

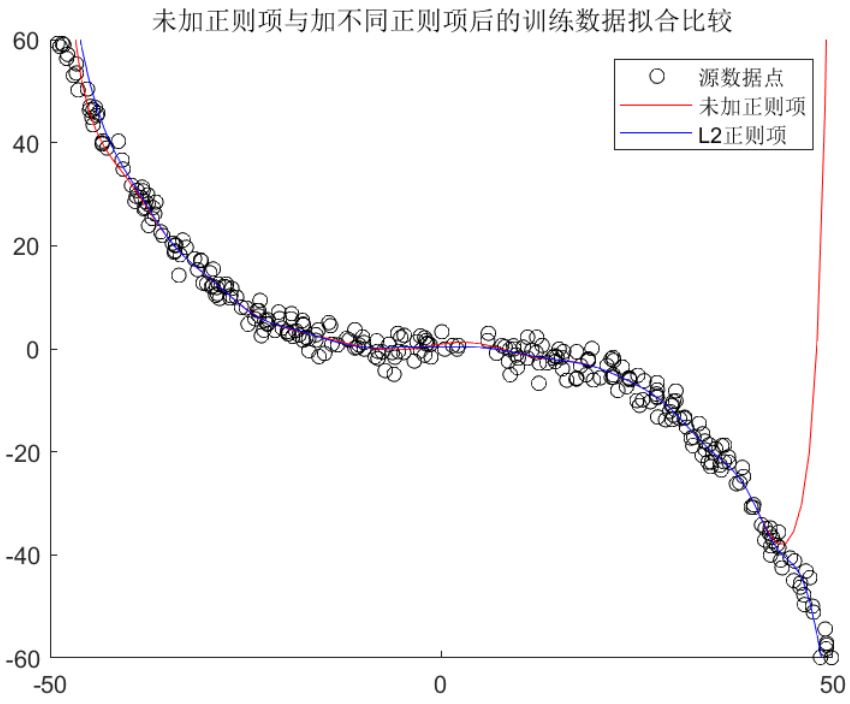

Figure 1: 300组训练数据集正则化前后拟合结果比较

在上方的300组训练数据集拟合过程中，可以明显地发现加L2正则后的数据图像比不加正则项的拟合图像有着更好的对应趋势，不加正则项的拟合图像在区间[40,50]附近已经明显偏离了原数据集。而加上L2正则项后趋势在区间[-50,50]之间一直保持较好。

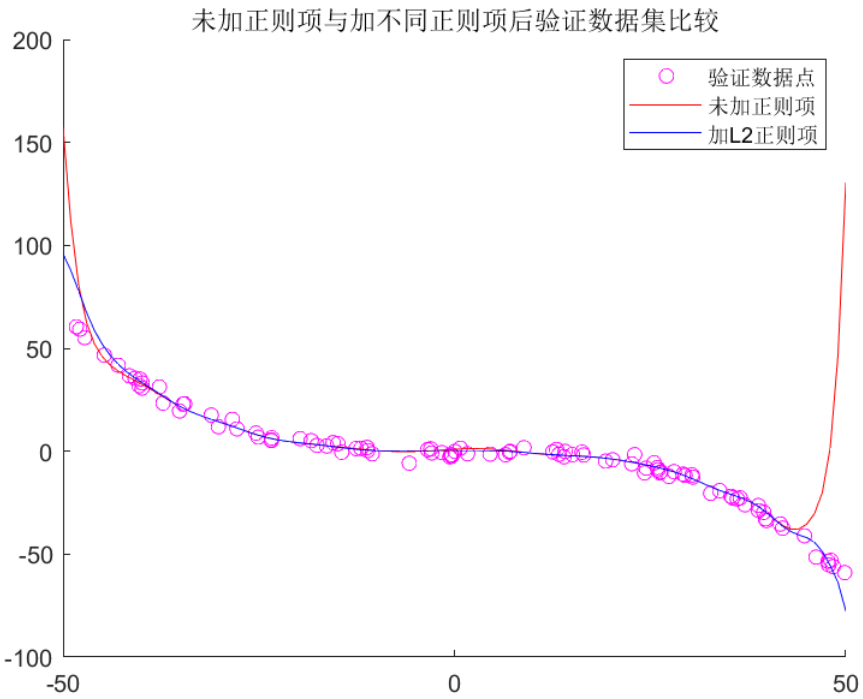

Figure 2: 100组验证数据集正则化前后拟合结果比较

而对于100组验证数据集的拟合，也可以发现与之前300组训练数据集的结果类似。这变相说明了加上L2正则项后的数据训练效果要明显好于不加正则项的直观结果。最后我们给出二者的误差用以作证这一事实(见下方误差图)

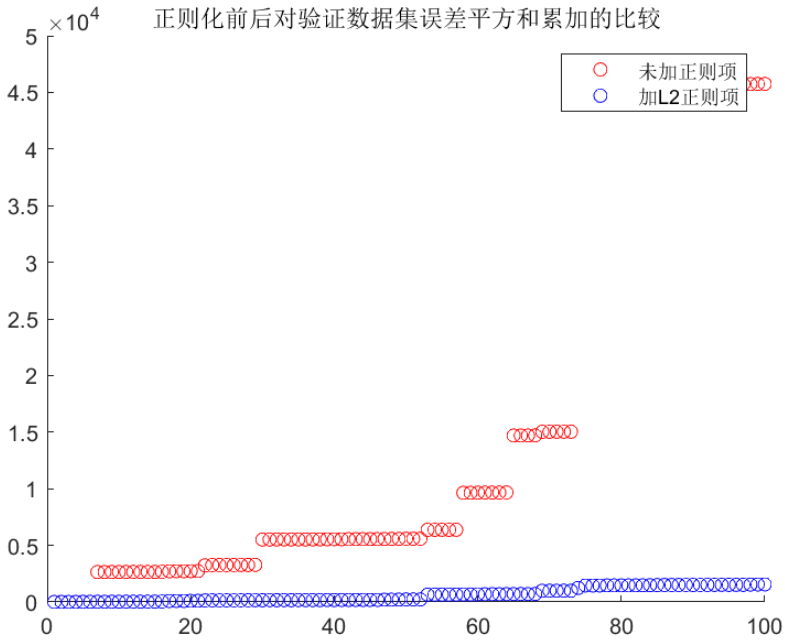

Figure 3: 100组验证数据集正则化前后误差序列比较

上述的实验中我选定了参数$\lambda = 10^6$，为了验证准确性，我修改参数$\lambda = 5 * 10^5$再一次实验，这次选择训练数据区间为[-40,40]，而验证数据区间为[-50,50]，这样更可以体现出数据的预测趋势。相关结果如下:

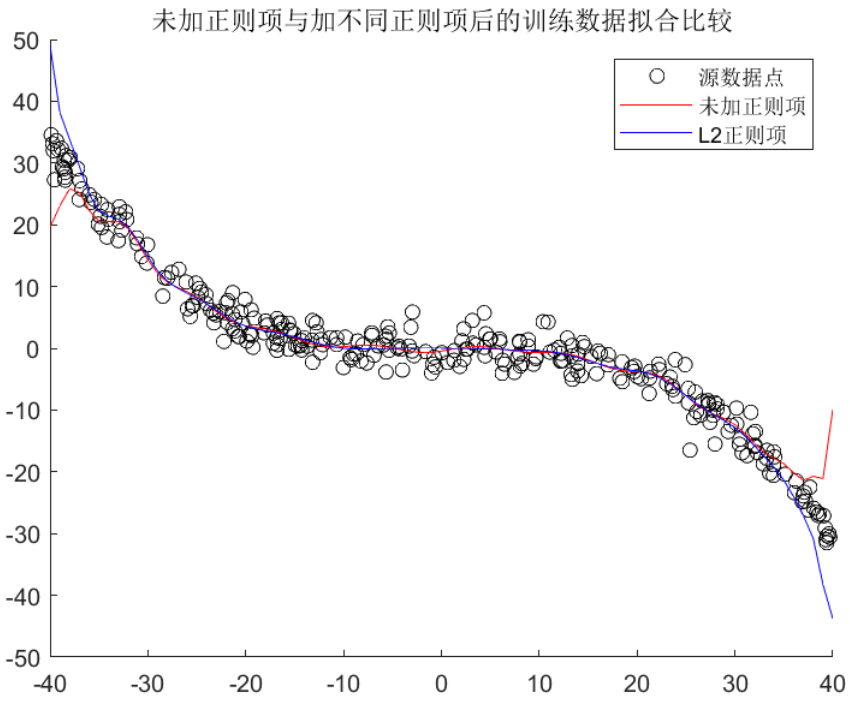

Figure 4: 300组训练数据集正则化前后拟合结果比较(2)

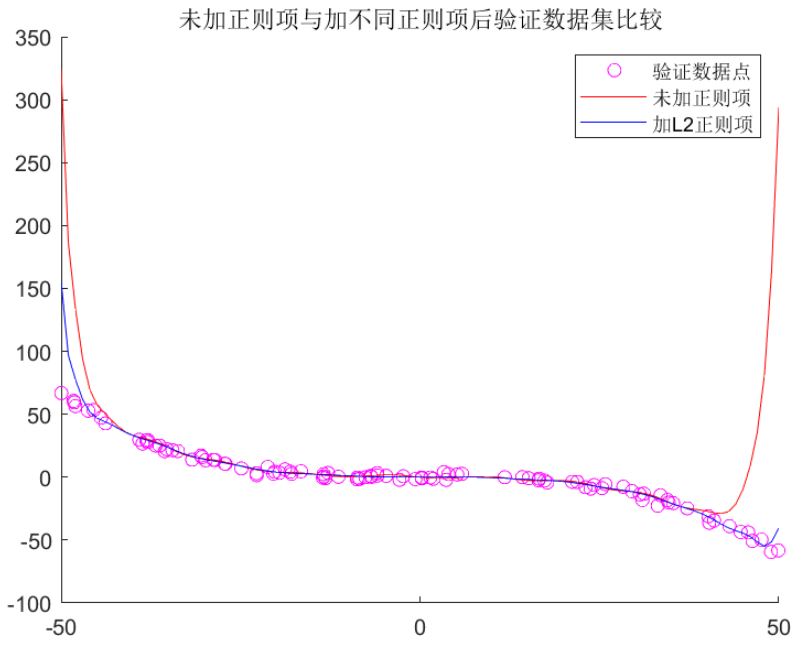

Figure 5: 100组验证数据集正则化前后拟合结果比较(2)

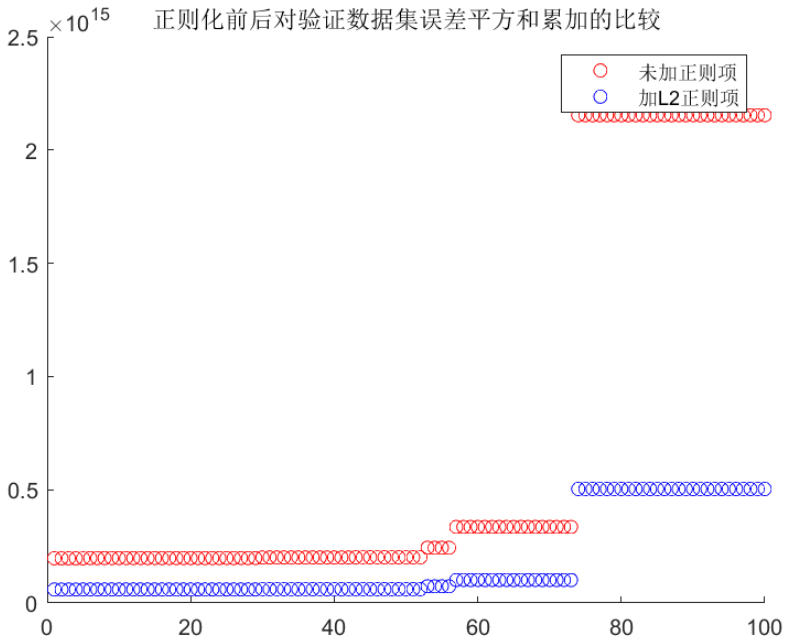

Figure 6: 100组验证数据集正则化前后误差序列比较(2)

发现将L2正则化前的超参数系数减小后，对比前后的误差差值相较以前会稍小一些，但是加上L2正则项后在改变验证区间的情况下，图像与点集仍旧近乎完全吻合，这肯定了L2范数正则化的作用，更有利于后续数据的进一步预测。

2 关于L1正则项

2.1 相关公式与近端梯度下降(PGD)

L1正则项范数就是价值向量各项绝对值之和的形式。加L1正则化表现度量:

$$\sum_{i=1}^{n}(f_w(x_i)-y_i)^2+\lambda*||w||_1 \tag{6}$$

但值得一提的是，对于L1正则化范数，因为涉及价值向量的绝对值，故在当价值向量存在分量为0时，此时对于价值向量w是不可导的，所以我们无法直接得到相应的解析解。因此我们需要动用近端梯度下降(PGD)的方法来对于梯度值以及对应的价值向量结果进行近似。

近端梯度下降法的思想在于将原表现度量$min_w\sum_{i=1}^{n}(f_w(x_i)-y_i)^2+\lambda*||w||_1$给拆分成两个式子

$$min_{w,z}\sum_{i=1}^{n}(f_w(x_i)-y_i)^2+\gamma*||w-z||_2 \tag{7}$$

与

$$min_z\gamma*||w-z||_2+\lambda*p(z) \tag{8}$$

先对后式进行处理得到对应z值再迭代入前式得到w值，若不满足终止条件就再代入后式。事实上，根据最优化理论，在迭代点x_k附近可以利用二阶Taylor展开

$$f(x)=f(x_k)+\nabla f(x_k)^{\mathrm{T}}(x-x_k)+\frac{L}{2}||x-x_k||_2=\frac{L}{2}||x-(x_k-\frac{1}{L}\nabla f(x_k))||_2^2+const \quad (9)$$

通过上式可得

$$x_{k+1}=x_k-\frac{1}{L}\nabla f(x_k) \quad (10)$$

接着借用神经网络BP算法中的梯度下降思想，利用梯度下降进行迭代，x_{k+1}理论值应为

$$x_{k+1}=argmin_x\frac{L}{2}||x-z||_2^2+\lambda||x||_1 \quad (11)$$

其中$z=x_k-\frac{1}{L}\nabla f(x_k)$ 依次迭代，可以得x的每一分量值

$$x_{k+1}^i=\begin{cases} z^i-\lambda/L & \lambda/L<z^i \\ 0 & |z^i|\le\lambda/L \\ z^i+\lambda/L & -\lambda/L>z^i \end{cases} \quad (12)$$

故可进行对于价值向量w的迭代逼近。

2.2 实验结果呈现与分析

在这里，我主要呈现L1与L2范数正则化的结果比较。我发现对于数量较多的数据集而言，L1范数正则达不到我想要的效果，因此我选择30个样本点的训练数据集以及10个样本点的验证数据集，并且做出了随着验证次数增加的相应的误差结果图(见下图)

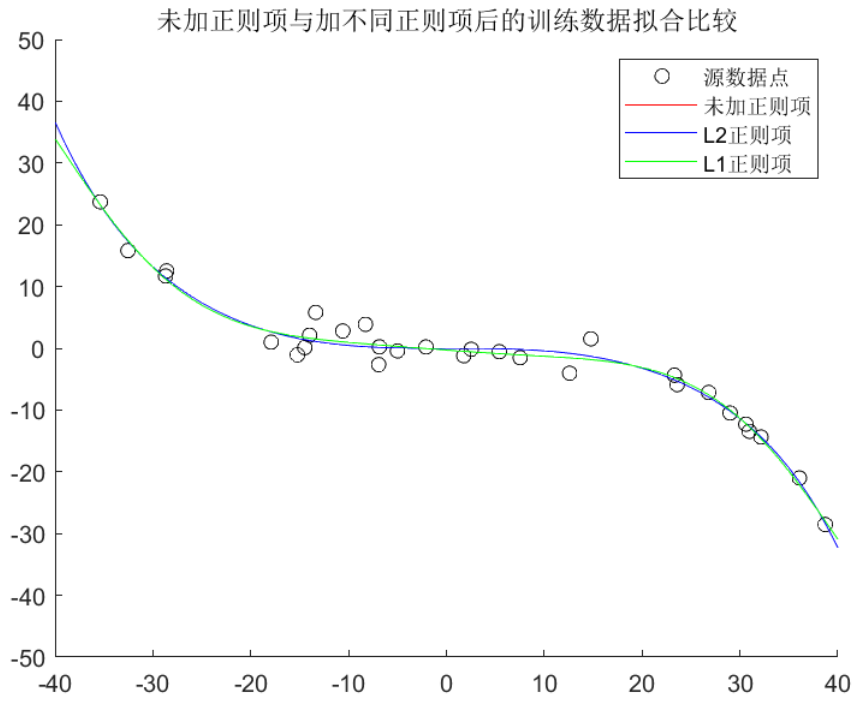

Figure 7: 30组训练数据集L1与L2正则拟合结果比较

从上图发现，对于30组训练数据集，二者的拟合图像近乎重合，但还是有着细微的变化，这可能和训练数据集样本数量不够大有一定关系。

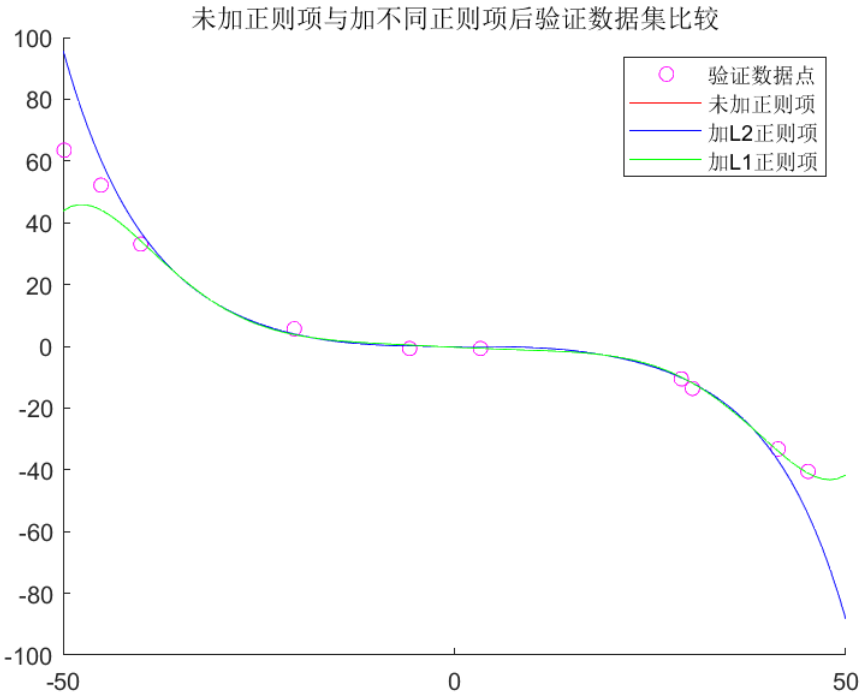

Figure 8: 10组验证数据集L1与L2正则拟合结果比较

从10组验证数据集的结果来看，发现L1正则化(上图中绿线)的拟合效果在直观上要比L2正则化(上图中蓝线)的效果要稍好一些，这主要是体现在了第一个和最后一个点上，绿线的L1正则有着明显靠近这两个点的趋势。最后，我们给出误差数据进行比较：

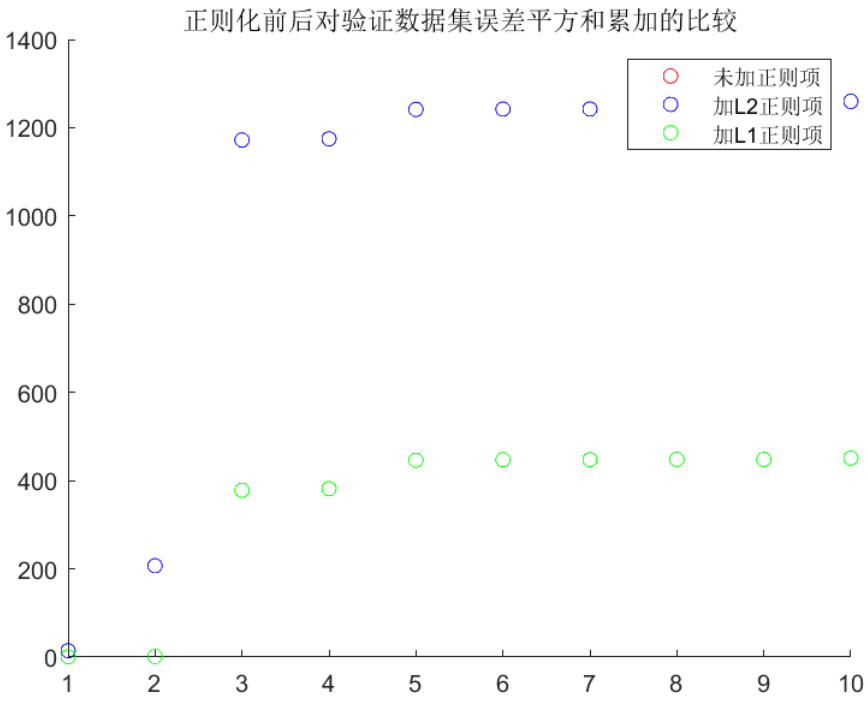

Figure 9: 10组验证数据集L1与L2正则拟合误差序列比较(2)

在这里，我做了将误差平凡和放大倍数的处理，发现L2正则化结果的误差要比L1正则化结果的误差要大一些。在这一组实验中，L1正则化的结果要稍好一些。

3 一些思考

不同正则项对于数据的回归拟合有着不同的效果。对于L1与L2两种范数的正则化，我的理解是对于表现度量中的价值向量的分量进行处理，从而达到减少或者抵消过拟合的效果。理论上的过拟合或者“硬”分类问题中，价值向量的选取往往会导致这些问题，从而使得数据曲线相当扭曲。L2正则化的效果在于将这些价值向量的分量乘上系数从而进行一定程度上的压缩，进而减小了所谓曲线的“陡峭程度”，使得曲线更为平缓，进而减小过拟合的程度，使得图像结果更加利于预测。而L1正则化的效果则在于可以将过拟合的价值向量中的一些项向零值逼近，也就是减小其绝对值大小，进而减小曲线的“陡峭程度”。值得一提的是，由于L1正则项的特殊性(绝对值函数)，进而其在平面上的表征为一正方形，故其极值更容易在坐标轴上取到，从而达到“稀疏化”的结果。事实上，要真正追求“稀疏化”的结果，最佳的选择是L0范数的正则化，但可惜的是，L0范数不是一个连续的函数表征，因此大大增加了偏导运算等等的难度，

所以现实中要达成稀疏化的结果，还是使用L1正则化的方法去进行不断逼近。总而言之，L2范数正则化倾向于价值向量的分量取值尽量均衡，即非零分量个数尽可能稠密，而L0和L1范数正则化则倾向于价值向量的分量尽量稀疏，集非零分量个数尽可能少。

事实上要解决过拟合问题，除了采用正则化方法，还可以选择早停的方式。在快要达到过拟合的临界阈值时，立刻终止训练，也是一个不错的方法。因此我们需要一个好的终止条件去进行“早停”的实验。

除此之外，肯定还会有更好的机器学习中解决过拟合的方法，这些都等待着发掘与探索。

4 相关代码

数据拟合正则化处理(不加正则项，L1正则，L2正则)的代码：(工具：matlab)

```
f=inline('(x.^4-1000*x.^3+5*x.^2-200*x+150)
    /(2*10.^6)','x');

a=-50:50;
b=f(a);
s_train=300;
s_val=100;
x_train=unifrnd(-50,50,1,s_train);
x_val=unifrnd(-50,50,1,s_val);
noiseintensity=2;%噪声强度
tnoise=noiseintensity*normrnd(0,1,1,s_train);
vnoise=noiseintensity*normrnd(0,1,1,s_val);
y_train=f(x_train)+tnoise;
y_val=f(x_val)+vnoise;
y_val0=zeros(1,s_val);
y_val2=zeros(1,s_val);
x_right=linspace(-50,50,0.1);
y_right=f(x_right);
figure(1);
hold on;
plot(x_train,y_train,'ko');
n=25;%拟合曲线的最高次数
x=[];
y=[];
for pow=1:size(x_train,2);
```

```
rx=[1];
for t=1:n-1
rx=[rx,x_train(1,pow).^t];
end
x=[x;rx];
y=[y;y_train(1,pow)];
end


lambda2=1000000;%设定正则化参数L2
e=eye(n,n);%设定单位矩阵
e(1,1)=0;
dim_p=size(x,2);%设定训练数据集的对应维数
u=ones(dim_p,1);
error0=0.0001;



ma=(((x')*x)^-1)*(x')*y;
ma_z=(((x')*x+lambda2*e)^-1)*(x')*y;
figure(1);
hold on;
i1=[-50:50];
j1=[0];
for t=1:n
j1=j1+ma(t).*(i1.^(t-1));
end
plot(i1,j1,'r');

i2=[-50:50];
j2=[0];
for t=1:n
j2=j2+ma_z(t).*(i2.^(t-1));
end
plot(i2,j2,'b');

axis([-50,50,-60,60]);
```

```
legend('源数据点','未加正则项','正则项L2');
title('未加正则项与加不同正则项后的训练数据拟合比较');

for t=1:n
y_val0=y_val0+ma(t).*(x_val.^(t-1));
end
for t=1:n
y_val2=y_val2+ma_z(t).*(x_val.^(t-1));
end
figure(2);
hold on;
plot(x_val,y_val,'mo');
plot(i1,j1,'r');
plot(i2,j2,'b');
legend('验证数据点','未加正则项','加正则项L2');
title('未加正则项与加不同正则项后验证数据集比较');

error0=zeros(1,100);
errorL2=zeros(1,100);
error0=y_val-y_val0;
error0_sum=zeros(1,100);
errorL2=y_val-y_val2;
errorL2_sum=zeros(1,100);
for t=1:100
error0_sum(t)=sumsqr(error0(1:t));
end
for t=1:100
errorL2_sum(t)=sumsqr(errorL2(1:t));
end

figure(3);
hold on;
time=[1:100];
plot(time,error0_sum,'ro');
plot(time,errorL2_sum,'bo');
legend('未加正则项','加正则项L2');
title('正则化前后对验证数据集误差平方和累加的比较');
```

```
f=inline('(x.^4-1000*x.^3+5*x.^2-200*x+150)
    /(2*10.^6)','x');

a=-50:50;
b=f(a);
s_train=30;
s_val=10;
x_train=unifrnd(-40,40,1,s_train);
x_val=unifrnd(-50,50,1,s_val);
noiseintensity=2;%噪声强度
tnoise=noiseintensity*normrnd(0,1,1,s_train);
vnoise=noiseintensity*normrnd(0,1,1,s_val);
y_train=f(x_train)+tnoise;
y_val=f(x_val)+vnoise;
y_val0=zeros(1,s_val);
y_val1=zeros(1,s_val);
y_val2=zeros(1,s_val);
x_right=linspace(-50,50,0.1);
y_right=f(x_right);
figure(1);
hold on;
plot(x_train,y_train,'ko');
n=8;%拟合曲线的最高次数
x=[];
y=[];
for pow=1:size(x_train,2)
rx=[1];
for t=1:n-1
rx=[rx,x_train(1,pow).^t];
end
x=[x;rx];
y=[y;y_train(1,pow)];
end
```

```
lambda1=10;%设定正则化参数L1
lambda2=1000000;%设定正则化参数L2
e=eye(n,n);%设定单位矩阵
e(1,1)=0;
dim_p=size(x,2);%设定训练数据集的对应维数
u=ones(dim_p,1);


ma=(((x')*x)^-1)*(x')*y;
ma_z=(((x')*x+lambda2*e)^-1)*(x')*y;
beta=0.5;
L=10;
w=ma;
wprev=ma;
prox_thres=lambda1*L;
totalerror=0.5;
w_tmp=zeros(n,1);
while true
grad_w=(x'*x)*w-x'*y;
z=w-grad_w*L;
for t=1:n
w_tmp(t,1)=sign(z(t))*max(abs(z(t))-prox_thres
    ,0);
end
w_diff=w-w_tmp;
km=y-x*w_tmp;
ko=y-x*w;
if sumsqr(w_diff)<totalerror
break
end
wprev=w;
w=w_tmp;
end
```

```
figure(1);
hold on;
i1=[-40:40];
j1=[0];
for t=1:n
j1=j1+ma(t).*(i1.^(t-1));
end
plot(i1,j1,'r');

i2=[-40:40];
j2=[0];
for t=1:n
j2=j2+ma_z(t).*(i2.^(t-1));
end
plot(i2,j2,'b');

i3=[-40:40];
j3=[0];
for t=1:n
j3=j3+w(t).*(i3.^(t-1));
end
plot(i3,j3,'g');

axis([-40,40,-50,50]);
legend('源数据点','未加正则项','正则项L2','正则
    项L1');
title('未加正则项与加不同正则项后的训练数据拟合比较');

for t=1:n
y_val0=y_val0+ma(t).*(x_val.^(t-1));
end
for t=1:n
y_val2=y_val2+ma_z(t).*(x_val.^(t-1));
```

```
end
for t=1:n
y_val1=y_val1+w(t).*(x_val.^(t-1));
end
i4=[-50:50];
j4=[0];
for t=1:n
j4=j4+ma(t).*(i4.^(t-1));
end

i5=[-50:50];
j5=[0];
for t=1:n
j5=j5+ma_z(t).*(i5.^(t-1));
end

i6=[-50:50];
j6=[0];
for t=1:n
j6=j6+w(t).*(i6.^(t-1));
end
figure(2);
hold on;
plot(x_val,y_val,'mo');
plot(i4,j4,'r');
plot(i5,j5,'b');
plot(i6,j6,'g');
legend('验证数据点','未加正则项','加正则项L2','加正则
    项L1');
title('未加正则项与加不同正则项后验证数据集比较');

error0=zeros(1,10);
errorL2=zeros(1,10);
errorL1=zeros(1,10);
error0=y_val-y_val0;
error0_sum=zeros(1,10);
errorL2=y_val-y_val2;
```

```
errorL2_sum=zeros(1,10);
errorL1=y_val-y_val1;
errorL1_sum=zeros(1,10);
for t=1:10
error0_sum(t)=sumsqr(error0(1:t));
end
for t=1:10
errorL2_sum(t)=sumsqr(errorL2(1:t));
end
for t=1:10
errorL1_sum(t)=sumsqr(errorL1(1:t));
end

figure(3);
hold on;
time=[1:10];
plot(time,error0_sum,'ro');
plot(time,errorL2_sum,'bo');
plot(time,errorL1_sum,'go');
legend('未加正则项','加正则项L2','加正则项L1');
title('正则化前后对验证数据集误差平方和累加的比较');
```