山东大学 计算机科学与技术 学院

机器学习 课程实验报告

|  |  |  |
| --- | --- | --- |
| 学号：201700301042 | 姓名：  陈佳睿 | 班级：  17智能 |
| 实验题目：正则化 | | |
| 实验过程中遇到和解决的问题：  （记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）  **一、正则化的线性回归**  本次实验包括两个集合的数据，一个用于线性回归，另外一个用于罗辑回归  实验指导书还提供了一个辅助函数map\_feature.m用于逻辑回归  正则化线性回归  输入数据x是一个独立特征 绘出的y可以看作是x在二维平面中的函数  使用matlab加载数据如下图所示  Screen Shot 2019-11-12 at 12.32.31 AM  可以看出只用一条直线去拟合数据太过简单了,效果也很差  Screen Shot 2019-11-12 at 10.19.26 AM  所以我们使用一个高阶多项式来更接近数据点  这里我们尝试五次多项式，所以等于我们假设了六个特征  我们要使用数据学习的分别是x^0, x^1,......x^5的系数  Screen Shot 2019-11-12 at 10.05.39 AM  尽管我们在使用多项式拟合数据  但是我们还是会在线性回归中出现问题  因为假设在每一个特征空间中是线性的  使用五次多项式对一个只有7个数据点的数据集进行拟合  不可避免的会导致过拟合问题  为了防止这一问题的出现，我们在模型中使用正则化来改善  在正则化问题中，我们的目标通过学习theta来最小化残差函数：  Screen Shot 2019-11-12 at 10.09.17 AM  上式中的 λ 就是正则化参数 正则化参数 λ 是对于我拟合参数的控制变量  如果随着我拟合参数的增大（即对于训练数据的拟合程度增大）  因为损失函数j（theta）中存在Screen Shot 2019-11-12 at 10.24.39 AM这一惩罚项，可以帮助我们减轻过拟合的现象  实验指导书提示我们：  notice that the summation after λ does not include （θ0）^2  那么这样我们就能找到对于模型拟合程度最好的标准方程  Screen Shot 2019-11-12 at 10.31.00 AM  Lamda = 0时的结果：  Screen Shot 2019-11-12 at 11.24.26 AM  Lamda = 1和Lamda = 10 的结果  Screen Shot 2019-11-12 at 2.17.39 PM  结论：可以看出，随着lamba的值的增大，拟合曲线越来越平滑，但是也并非越平滑越符合训练数据的形式，相比lamba=1和lamda=10，在这7个数据的训练集上，lamda=0的情况要更加符合数据的表达。  当然，这也并不代表在测试集上模型会有很好的表现  所以调参也存在一定的运气成分，当然一个理想的可解释的机器学习模型才是我们调参数的基础  对比实验指导书所给出的结果图，可以认为基本与其预期效果一致  Screen Shot 2019-11-12 at 2.32.44 PM   1. **正则化的逻辑回归**   实验的第二部分是要我们使用牛顿法来实现正则化的逻辑回归  首先，在matlab中加载数据  得到的效果如下图所示：  Screen Shot 2019-11-12 at 2.49.41 PM  在逻辑回归中，我们使用的假设函数：  将任意输入值限制在[0,1]之间  Screen Shot 2019-11-12 at 3.19.35 PM  加入正则项后，我们的损失函数表示如下：  Screen Shot 2019-11-12 at 3.36.42 PM  我们将x赋值为u和v的所有单项式(即多项式项)直到六次幂  即将我们的数据特征映射为27个维度  再加上一个阈值项一共28个维度作为数据的特征  Screen Shot 2019-11-12 at 7.52.57 PM  这里我们使用实验指导书中给出的map\_features函数  来进行向高维度的特征映射  再在实现的公式中加入正则项  那么新的目标函数形式如下：  Screen Shot 2019-12-03 at 3.33.17 PM  因为我们无法得到这个问题的解析解  所以我们使用梯度下降的方法来进行求解  Screen Shot 2019-11-12 at 8.03.10 PM  目标函数对theta进行求导数得到如下的表达  Screen Shot 2019-11-12 at 8.04.04 PM  Hessian矩阵形式  Screen Shot 2019-11-12 at 8.04.44 PM  将梯度变形：  Screen Shot 2019-11-12 at 8.28.46 PM  **3.实验结果分析**  1) lamda = 0  Screen Shot 2019-12-03 at 4.55.13 PM   1. lamda = 1   Screen Shot 2019-12-03 at 5.00.26 PM  3) lamda=10  Screen Shot 2019-12-03 at 5.03.34 PM | | |
| 结果分析与体会：通过这次实验，通过对线性回归和逻辑回归使用正则化，可以看到借助在目标函数加上不同表示的正则化项（惩罚项）可以很好地达到避免过拟合的效果。 | | |

代码：

x = load ('ex5Logx.dat');

y = load ('ex5Logy.dat');

figure

% Find the indices for the 2 classes

pos = find(y);

neg = find(y == 0);

plot(x(pos, 1), x(pos, 2), '+');

hold on

plot(x(neg, 1), x(neg, 2),'o');

xlabel('data1');

ylabel('data2');

I=eye(28);

I(1,1)=0;

m=length(y);

%å¾—åˆ°28ç»´çš„Xç”¨äºŽè®¡ç®—boundary

X=map\_feature(x(:,1),x(:,2));

%% regulization

% è®¾ç½®æœ€å¤§å¾ªçŽ¯æ¬¡æ•°

num=500;

% %% lamda=0        %% 1 10

% lamda=0;

% theta=zeros(28,1);

% for j=1:num

%     % hæ˜¯117\*1ç»´

%     h=1./(1+exp(-X\*theta));

%     % X=117\*28ç»´ theta=28\*1 Xâ€˜=28\*117ç»´

%     H=(1/m)\*(X'\*(h.\*(1-h).\*X)+lamda\*I);

%     invH=inv(H);

%     theta=theta-invH\*(1/m)\*(X'\*(h-y)+lamda\*I\*theta);

% end

%

%

% % plot

% u=linspace(-1,1.5,200);

% v=linspace(-1,1.5,200);

% z=zeros(length(u),length(v));

% for j=1:length(u)

%     for k=1:length(v)

%         z(j,k)=map\_feature(u(j),v(k))\*theta;

%     end

% end

% hold on

% contour(u,v,z',[0,0],'linewidth',1);

% %% lamda=1

% lamda=1;

% theta=zeros(28,1);

% for j=1:num

%     h=1./(1+exp(-X\*theta));

%     H=(1/m)\*(X'\*(h.\*(1-h).\*X)+lamda\*I);

%     invH=inv(H);

%     theta=theta-invH\*(1/m)\*(X'\*(h-y)+lamda\*I\*theta);

% end

% % plot

% u=linspace(-1,1.5,200);

% v=linspace(-1,1.5,200);

% z=zeros(length(u),length(v));

% for j=1:length(u)

%     for k=1:length(v)

%         z(j,k)=map\_feature(u(j),v(k))\*theta;

%     end

% end

% hold on

% contour(u,v,z',[0,0],'linewidth',2);

 %% lamda=10

lamda=10;

theta=zeros(28,1);

for j=1:num

    h=1./(1+exp(-X\*theta));

    H=(1/m)\*(X'\*(h.\*(1-h).\*X)+lamda\*I);

    invH=inv(H);

    theta=theta-invH\*(1/m)\*(X'\*(h-y)+lamda\*I\*theta);

end

% plot

u=linspace(-1,1.5,200);

v=linspace(-1,1.5,200);

z=zeros(length(u),length(v));

for j=1:length(u)

    for k=1:length(v)

    z(j,k)=map\_feature(u(j),v(k))\*theta;

    end

end

hold on

contour(u,v,z',[0,0],'linewidth',4);

% legend('positive','negative','lamda=0','lamda=1','lamda=10');