

**模式识别大作业**

学 院 信息科学与工程

专 业 控制科学与工程

组 员 徐逸峰

指导教师 赵海涛

**完成日期： 2018 年 10 月22日**

**模式识别作业报告**

组员：徐逸峰 Y30180684

经过若干次模式识别课程的学习和多次赵老师的指导，我对于最小二乘法、梯度下降法、SVD、PCA、朴素贝叶斯、Logistic回归等理论以及其数学原理有了初步的了解，本次模式识别作业采用的方法为Logistic回归，以巩固对于该算法的学习。

1. Criteo 展示广告

作业选题为lintcode人工智能题目中的“Criteo展示广告”， 本题我们使用Criteo所共享的一周展示广告数据，数据中提炼了13个连续特征、26个离散特征和用户是否点击了该页面广告的标签。预测用户在不同的特征下是否会点击广告。本次作业仅选用其中12个连续特征利用Logistic回归进行预测学习。

1.1 算法简介

Logistic Regression的整个过程就是在面对一个回归或者分类问题时，采用建立代价函数，然后通过优化方法迭代求解出最优的模型参数，然后测试验证我们这个求解的模型的好坏.在回归模型中，通常y是一个定性变量，比如y=0或1，logistic方法主要应用于研究某些事件发生的概率或进行某些事物的二分类。

1.2 算法步骤

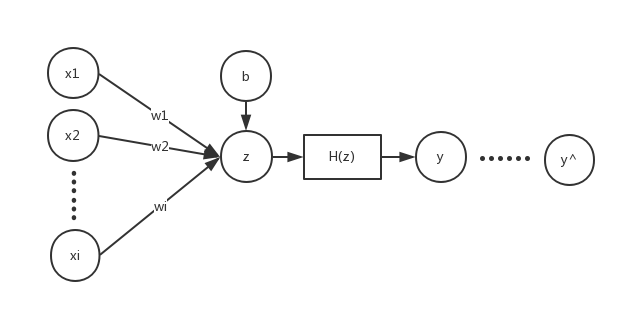
Regression的常规步骤为以下三步：

1）寻找hypothesis Function（即预测函数）；

2）构造Loss Function ；

3）想办法使得Loss Function函数最小并求得回归参数集θ。

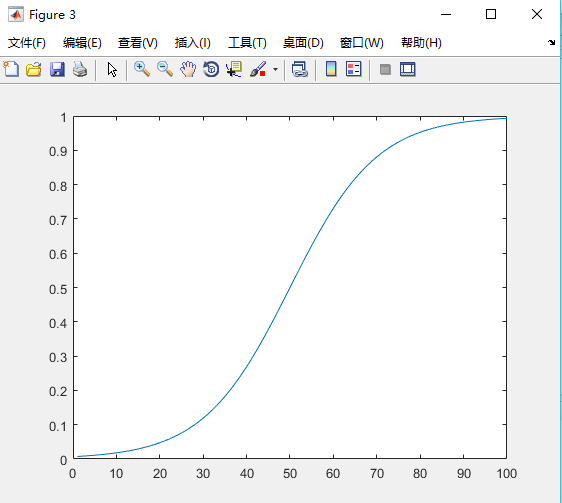
1.2.1 Logistic回归模型

本次作业模型采用线性模型，结构如图(1-1)所示

**图1-1 Logistic回归数据过程示意图**

1.2.2预测函数

Logistic回归算法主要应用与二分类问题以及概率预测问题的解决，所以是用了sigmiod函数作为其预测函数，其函数形式为：

该函数拥有一个漂亮的s型形状，且单调递增、值域为(0,1)，函数图像如图(1-2)所示。

**图1-2 sigmoid函数示意图**

1.2.3构造Loss Function

下式表示的是最后结果取1的概率，因此对输入x分类结果为1和0的概率分别为:

,

表达于一个式子中即为：

因此Loss Function取其似然函数为：

其中m为数据样本点数，实际求解过程中使用梯度下降求取Loss Function的最小值，因此对求对数再乘以得到最后的Loss Function为：

1.2.4求解Loss为最小值时的参数集

采用梯度下降法进行求解，将Loss函数对求偏导数。

由于以及，根据求取偏微分的链式法则可知：

同理可得：=

因此：

对上述公式进行向量化，其中在特征矩阵X中添加全1列以构造参数b。

Sigmiod自变量，;

损失函数;

梯度;

参数集B迭代公式为.其中a为learning rate。

1.3解决方案

1.3.1数据导入

采用Matlab中的csvread函数对训练集数据‘train.csv’进行导入，本次作业仅对数据特征(I类)进行训练学习，发现各个特征都有部分值缺失，其中I12缺失了绝大部分值，因此不作为训练特征使用。

1.3.2数据补全

由于数据缺失，需要对各个特征数据进行补全，编写了数据补全matlab函数completion.m如下所示，将各个特征中的缺失值使用平均值，众数或0补全（实际情况中众数与0补全效果差距不大）。

1.3.3特征缩放

由于各个特征所处的值域范围不同，为了确保这些不同的特征能够处在一个相近的范围内，使梯度下降法迭代过程能更快地收敛，因此需要对各个特征进行特征缩放。其数学原理为；

其中为缩放后的特征值，为特征平均值，为特征标准差。

编写Matlab函数Feascaling.m如下所示：

%数据特征缩放

%输入P为需要进行特征缩放的数据列矢量

%输出P1为特征缩放后的数据列矢量

function [P1]=Feascaling(P)

[m,n]=size(P);

Pmean=mean(P);%均值

Pstd=std(P);%标准差

for i=1:m

P1(i,:)=(P(i,:)-Pmean)/(Pstd);%特征缩放

end

1.3.4主函数与正则化

为防止模型过拟合使得模型泛化能力差，进行正则化，即在Loss Function中添加正则化项得到新的损失函数：

主函数adv\_logistic\_gradientd.m如下所示：

%广告点击预测 Logistic回归 I12缺失数据过多不作为特征

%数据补全

%conpletion(A,k)A为补全数据集，k为补全模式：1平均值 2众数

I1=completion(I1,2);

I2=completion(I2,2);

I3=completion(I3,2);

I4=completion(I4,2);

I5=completion(I5,2);

I6=completion(I6,2);

I7=completion(I7,2);

I8=completion(I8,2);

I9=completion(I9,2);

I10=completion(I10,2);

I11=completion(I11,2);

I13=completion(I13,2);

%数据特征处理

%对数据特征进行特征缩放

I1=Feascaling(I1);

I2=Feascaling(I2);

I3=Feascaling(I3);

I4=Feascaling(I4);

I5=Feascaling(I5);

I6=Feascaling(I6);

I7=Feascaling(I7);

I8=Feascaling(I8);

I9=Feascaling(I9);

I10=Feascaling(I10);

I11=Feascaling(I11);

I13=Feascaling(I13);

A=[I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I13];

[m,dim]=size(A);%特征维度

for i=1:m

A(i,dim+1)=1;

end

X=A(:,1:dim+1);%训练集数据

Y=Label;%训练集label

B=zeros(dim+1,1);%初始化参数矩阵

step=0;%迭代步数

Z=X\*B;

for j=1:m

H(j,:)=1/(1+exp(-Z(j,:)));%sigmiod函数

end

E(1,:)=(-1/m)\*(Y'\*log(H)+(1-Y')\*log(1-H));

J=X'\*(H-Y)/m;

a=0.03;%learning rate

lambda=10;%正则化系数

while step<6000

sum=0;%正则化项

Z=X\*B;%simoid自变量 m\*1维

step=step+1;

for j=1:m

H(j,:)=1/(1+exp(-Z(j,:)));%sigmiod函数

end

I=eyes(dim+1,dim+1);

for j=1:dim

sum=sum+B(j,:)\*B(j,:);

end

EC(step,:)=lambda\*sum/m;

E(step,:)=(-1/m)\*(Y'\*log(H)+(1-Y')\*log(1-H))+lambda\*sum/m;%Loss Function

J=X'\*(H-Y)/m+lambda\*B/m;%梯度

B=B-a\*J;%梯度迭代

end

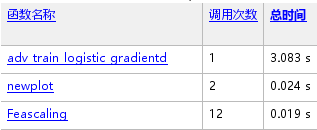
figure(1);

plot(E);%绘制loss与迭代次数的关系图

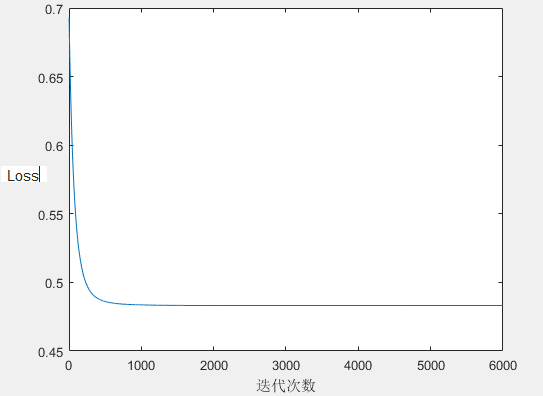
figure(2);

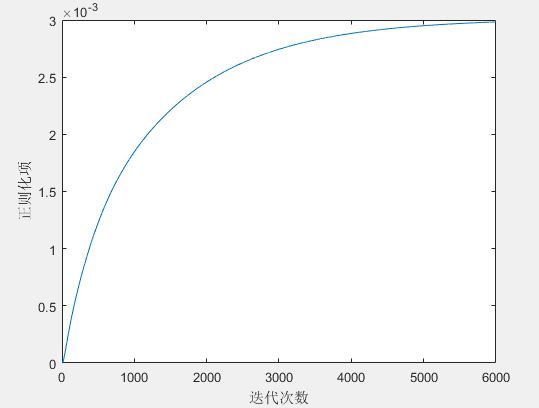
plot(EC);%绘制正则化项与迭代次数的关系图

1.3.4训练函数运行结果

一些主要函数运行耗时如图(1-3)所示，总时间不到十秒。

**图1-3 各函数运行时间**

损失函数值以及正则化项分别于迭代次数的关系图如图(1-4)、图(1-5)所示，可以发现正则化项不到最终Loss值的1%。而Loss值在2000次迭代过后已经进入稳态。

**图1-4 Loss值和迭代次数的关系**

**图1-5 正则化项和迭代次数的关系图**

1.3.5测试函数

编写测试函数adv\_test.m如下所示：

%广告点击预测test

%数据补全

%conpletion(A,k)A为补全数据集，k为补全模式：1平均值 2众数

I1=completion(I1,2);

I2=completion(I2,2);

I3=completion(I3,2);

I4=completion(I4,2);

I5=completion(I5,2);

I6=completion(I6,2);

I7=completion(I7,2);

I8=completion(I8,2);

I9=completion(I9,2);

I10=completion(I10,2);

I11=completion(I11,2);

I13=completion(I13,2);

% %数据特征处理

I1=Feascaling(I1);%特征缩放

I2=Feascaling(I2);

I3=Feascaling(I3);

I4=Feascaling(I4);

I5=Feascaling(I5);

I6=Feascaling(I6);

I7=Feascaling(I7);

I8=Feascaling(I8);

I9=Feascaling(I9);

I10=Feascaling(I10);

I11=Feascaling(I11);

I13=Feascaling(I13);

dim=12;%特征维度

A=[I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I13];

[m,n]=size(A);

for i=1:m

A(i,dim+1)=1;

end

X=A;

Z=X\*B;%simoid自变量 m\*1

for j=1:m

H(j,:)=1/(1+exp(-Z(j,:)));%激励函数

end

submission=[Id,H];

csvwrite('advcsv11.csv',submission);

1.3.6结果提交

通过csvwrite函数将结果输出为csv文件并在lincode上提交，结果如图(1-6)所示。

**图1-6 广告点击预测结果**

2泰坦尼克号生存预测

本题选用Age,Fare,Parch,Pclass,Sex,SibSp,Embarked7维特征对模型进行学习。其中Pclass : 乘客的船票的等级；Sex : 乘客性别(male, female)；Age : 乘客年龄；Sibsp ：船上兄弟姐妹/配偶的人数；Parch : 船上父母/儿女的人数；Ticket : 船票号码；Fare : 船票价格；Embarked : 出发港口。

本题依旧采用Logistic回归和梯度下降法完成该问题的预测，由于Logistic回归的基本原理等前面已经提及，此处不做赘述。依旧使用sigmoid函数，以及原来的Loss作为损失函数。利用梯度下降法求出参数集。

2.1解决方案

2.1.1数据导入

利用csvread函数读取训练集‘train.csv’，选取已经提及的7维特征，暂时不区分数据特征与类别特征均作为数据特征处理。

2.1.2数据预处理

使用前面已经提到的‘completion.m’,与‘Feascaling.m’对数据进行预处理，这里由于只有年龄和船费和其他数据值域范围差距过大，因此只对这两个特征进行特征缩放。

2.1.2训练函数

训练主函数如下所示：

%泰坦尼克号生存预测 Logistic回归 梯度下降

%特征数据预处理

Age=completion(Age,1);

Fare=completion(Fare,1);

Embarked=completion(Embarked,2);

%类别特征处理

% Pclass=classcode(Pclass,3);%对类别特征进行编码

% Sex=classcode(Sex,2);

% Embarked=classcode(Embarked,3);

A=[Age,Fare,Parch,Pclass,Sex,SibSp,Embarked];

[m,dim]=size(A);%dim特征维度 m数据维数

for i=1:m

A(i,dim+1)=1;

end

A=[A,Survived];

X=A(:,1:dim+1);%训练集

Y=A(:,dim+2);%测试集的label

B=zeros(dim+1,1);%初始化参数矩阵

step=0;%迭代步数

a=0.002;%learning rate

while step<40000

Z=X\*B;%simoid自变量

step=step+1;

for j=1:m

H(j,:)=1/(1+exp(-Z(j,:)));%sigmiod函数

end

J=X'\*(H-Y)/m;%梯度

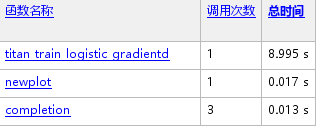
B=B-a\*J;%梯度下降

end

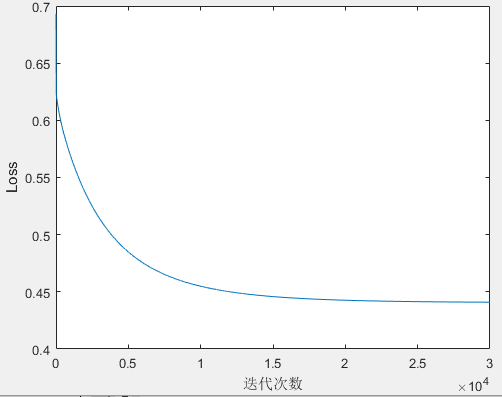
figure(1);%绘制Loss与迭代次数的关系图

plot(E);

2.1.3训练函数运行结果

训练函数中各个函数的主要运行时间如图(2-1)所示，总运行时间不到10秒。

**图2-1 泰坦尼克生存预测训练函数运行时间**

Loss值与迭代次数的关系如图(2-2)所示(学习率为0.003，实际调试时学习率超过0.005会出现loss值变大，无法收敛至最小值，因此取0.003)，可以看出25000步后基本收敛至最小值。

**图2-2 Loss与迭代次数的关系**

2.1.4 测试函数运行

测试函数如下所示：

%数值特征处理

Age=completion(Age,3); Fare=completion(Fare,3);

Embarked=completion(Embarked,2);

Age=Feascaling(Age);Fare=Feascaling(Fare);

A=[Age,Fare,Parch,Pclass,Sex,SibSp,Embarked];

[m,dim]=size(A); X=A(:,1:dim);

for i=1:m

X(i,dim+1)=1;

end

Z=X\*B;%simoid自变量 m\*1

for j=1:m

H(j,:)=1/(1+exp(-Z(j,:)));%激励函数

if H(j,:) >= 0.5

Y1(j,:)=1;

else

Y1(j,:)=0;

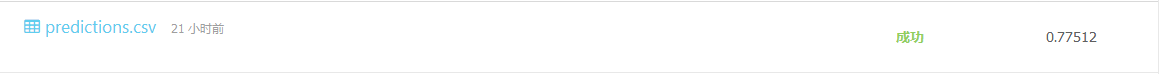
end

end

submission=[PassengerId,Y1];

2.1.5 测试结果提交

通过csvwrite函数将结果输出为csv文件在lintcode网站上提交结果如图(2-3)所示，而最小二乘法预测出的结果如图(2-4)所示。明显发现Logistic回归的预测结果没有达到最小二乘法直接计算得到的最优解的效果，暂时还没有理解其中原因。

**图2-3 泰坦尼克生存预测Logistic回归预测结果**

**图2-4 泰坦尼克生存预测最小二乘预测结果**

3三作业总结

通过两道题目的实际实践，更加熟悉了Logistic回归算法，对于编程也更加熟练。同时在实际过程中出现了一些暂时没有着手解决的问题。例如，对于类别特征与数据特征都纳入计算的泰坦尼克生存预测问题中，尝试使用1-of-k编码的方式对每个类别特征都扩充成k\*m(m为样本点数)的矩阵，结果发现正确率反而下降。此外，由于对年龄的缺失值都填充平均值或是众数会影响原本年龄的分布，尝试建立了一个新的模型，用其余6维特征作为训练年龄的特征，有年龄的样本做训练集，年龄缺失的样本做测试集，对年龄进行预测，最后结果正确率也是下降了。后续需要查询并学习更多的相关资料与知识，以解决目前的困惑。

这次的作业进一步增加了了我对机器学习、模式识别的兴趣，并对本来立足于想象中的算法付诸实践，使自己对其更为熟练。感谢赵海涛老师上课的仔细教学，以及课外的解答困惑。

附：github中文件说明

Feascaling.m 数据特征缩放函数

completion.m 特征补全函数

adv\_train\_logistic\_gradientd.m 广告点击预测训练函数

adv\_test.m 广告点击预测测试函数

advcsv.csv 广告点击预测结果

titan\_train\_logistic\_gradientd.m 泰坦尼克生存预测训练函数

titan\_test.m 泰坦尼克生存预测测试函数

titan\_csv.csv 泰坦尼克生存预测结果