哈尔滨工业大学计算机科学与技术学院

实验报告

课程名称： 机器学习

课程类型： 选修

实验题目： logistic 回归

学号：1161800218

姓名： 陈翔

一、实验目的

理解逻辑回归模型，掌握逻辑回归模型的参数估计算法。

二、实验要求及实验环境

实验要求：

实现两种损失函数的参数估计（1，无惩罚项；2.加入对参数的惩罚），可以采用梯度下降、共轭梯度或者牛顿法等。
验证：1.可以手工生成两个分别类别数据（可以用高斯分布），验证你的算法。考察类条件分布不满足朴素贝叶斯假设，会得到什么样的结果。
2. 逻辑回归有广泛的用处，例如广告预测。可以到 UCI 网站上，找一实际数据加以测试。

实验环境：

win10，64 位操作系统，MATLAB

三、设计思想（本程序中的用到的主要算法及数据结构）

1．算法原理

logistic 回归主要是进行二分类预测，也即是对于 0~1 之间的概率值，当概率大于 0.5 预测为 1，小于 0.5 预测为 0. 通过函数映射的方法，把数据拟合的结果映射到 0 或 1 上。
要把拟合结果映射到 0,1 上来，那映射函数的取值应该只有 0 和 1 两种，首先想到的是阶跃函数

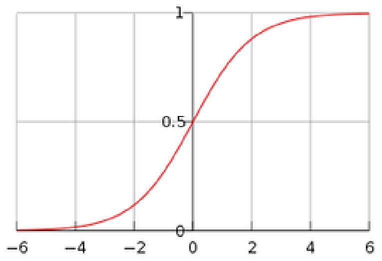
$$y=\begin{cases}1, & \text{if } x>0\\ 0, & \text{if } x\leqslant 0\end{cases}$$

但是问题在于这个函数在阶跃点上从 0 瞬间变为 1，很难处理。
sigmoid 函数具有相似的性质，并且在数学上更容易处理

$$\sigma(z)=\frac{1}{1+\mathrm{e}^{-z}}$$

当 x 为 0 时，sigmoid 函数值为 0.5，随着 x 的增大，对应的 sigmoid 函数值逼近 1，随着 x 的减小，sigmoid 函数值逼近 0.

$$f(z)=\frac{e^z}{e^z+1}=\frac{1}{1+e^{-z}},$$



因此，为了实现 logistic 分类器，我们可以在每个特征上都乘以一个回归系数，

然后所有的相乘结果进行累加，将这个总结作为输入，输入到 sigmoid 函数中，从而得到一个大小为 0~1 之间的值，当该值大于 0.5 归类为 1，否则归类为 0，这样就完成了二分类的任务。所以 logistic 回归可以看成是一种概率估计。

$$z = w_0x_0 + w_1x_1 + w_2x_2 + \cdots \qquad W^{\mathrm{T}}X$$

向量 x 是特征变量，是输入数据。此数据有两个特征，可以表示为

$$z = w_0x_0 + w_1x_1 + w_2x_2$$

w_0 是常数项，需要构造 x_0 等于 1。向量 W 是回归系数特征，之后就是确定最佳回归系数 $W(w_0, w_1, w_2)$。

逻辑斯谛回归模型学习时，对于给定的训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，其中，$x_i\in\mathbf{R}^n$，$y_i\in\{0,1\}$，可以应用极大似然估计法估计模型参数，从而得到逻辑斯谛回归模型.

设： $P(Y=1\mid x)=\pi(x)$， $P(Y=0\mid x)=1-\pi(x)$

似然函数为

$$\prod_{i=1}^{N}[\pi(x_i)]^{y_i}[1-\pi(x_i)]^{1-y_i}$$

此为联合概率分布函数

对数似然函数为

$$\begin{aligned}
L(w) &= \sum_{i=1}^{N}[y_i\log\pi(x_i)+(1-y_i)\log(1-\pi(x_i))] \\
&= \sum_{i=1}^{N}\left[y_i\log\frac{\pi(x_i)}{1-\pi(x_i)}+\log(1-\pi(x_i))\right] \\
&= \sum_{i=1}^{N}[y_i(w\cdot x_i)-\log(1+\exp(w\cdot x_i)]
\end{aligned}$$

对 $L(w)$ 求极大值，得到 w 的估计值.

这样，问题就变成了以对数似然函数为目标函数的最优化问题．逻辑斯谛回归学习中通常采用的方法是梯度下降法及拟牛顿法.

分类函数为 $h(z)=\dfrac{1}{1+\mathrm{e}^{-z}}$，因此预测函数为 $h(z)=h_w(x)=\dfrac{1}{1+\mathrm{e}^{-W^{\mathrm{T}}X}}$

$$P(y=1\mid x;\theta)=h_\theta(x)$$
$$P(y=0\mid x;\theta)=1-h_\theta(x)$$

$$\cos\mathrm{t}(h_\theta(x),y)=\begin{cases}-\log(h_\theta(x)), & \text{if } y=1\\ -\log(1-h_\theta(x)), & \text{if } y=0\end{cases}$$

$$J(\theta)=\frac{1}{m}\sum_{i=0}^{m}\cos\mathrm{t}\left(h_\theta\left(x^{(i)}\right),y^{(i)}\right)=-\frac{1}{m}\left[\sum_{i=0}^{m}y^{(i)}\log h_\theta\left(x^{(i)}\right)+\left(1-y^{(i)}\right)\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right]$$

于是，将求得的偏导代入梯度上升法迭代公式

$$w_j := w_j+\alpha\sum_{i=1}^{m}\left(y^{(i)}-h_w\left(x^{(i)}\right)\right)x_j^{(i)}$$

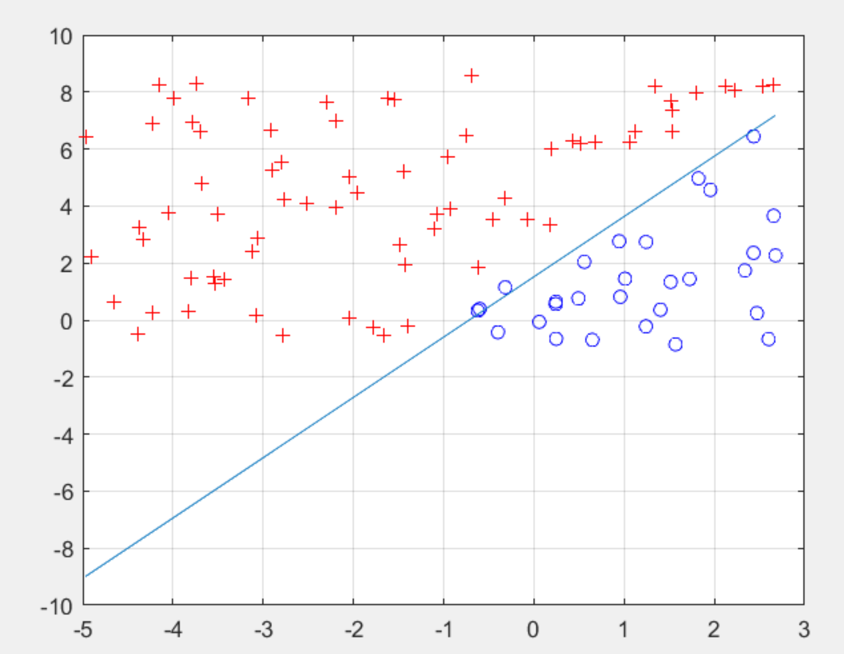
2. 算法的实现

先手工随机生成两个分别类别数据
然后根据以上算法流程得到 $W(w_0, w_1, w_2)$

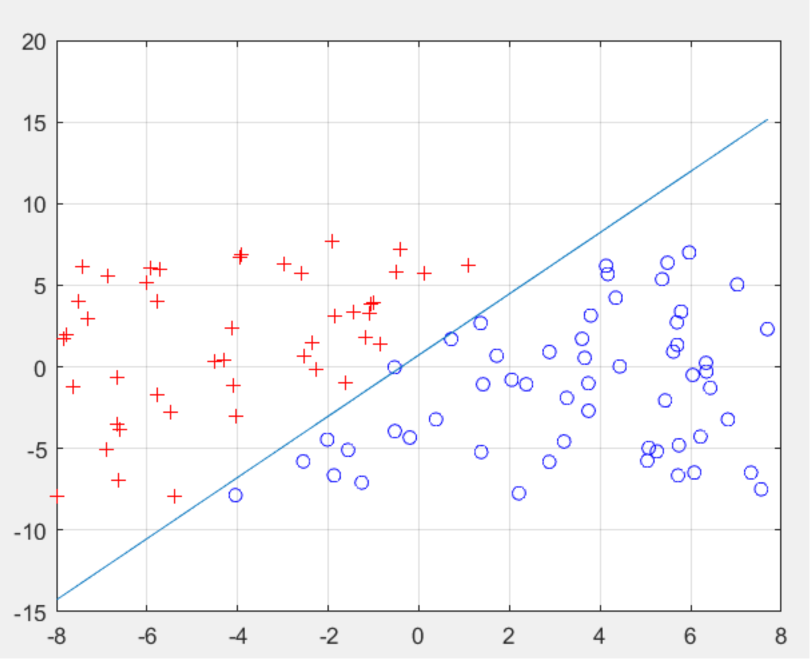
由于之前预测的直线方程 $0 = w_0x_0 + w_1x_1 + w_2x_2$， 带入回归系数，可以确定边界

$$x_2 = -(w_0 + w_1x_1) / w_2$$

四、实验结果与分析

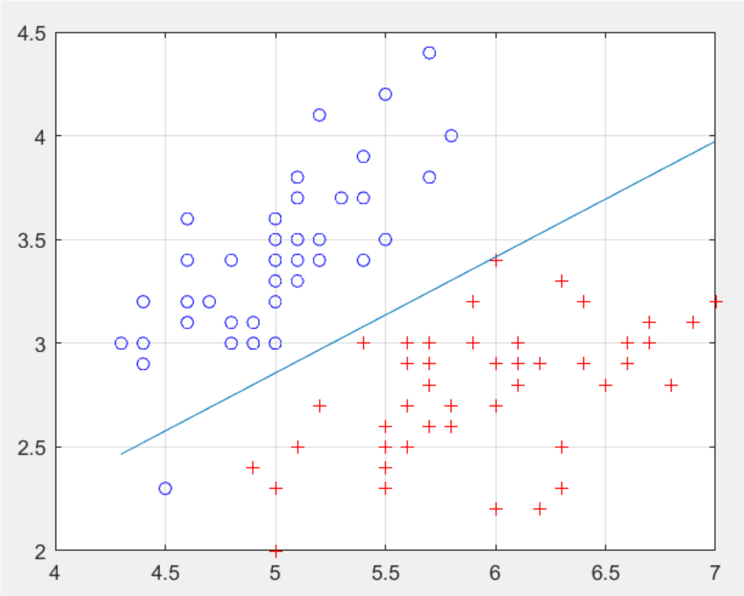


如果类条件分布不满足朴素贝叶斯假设



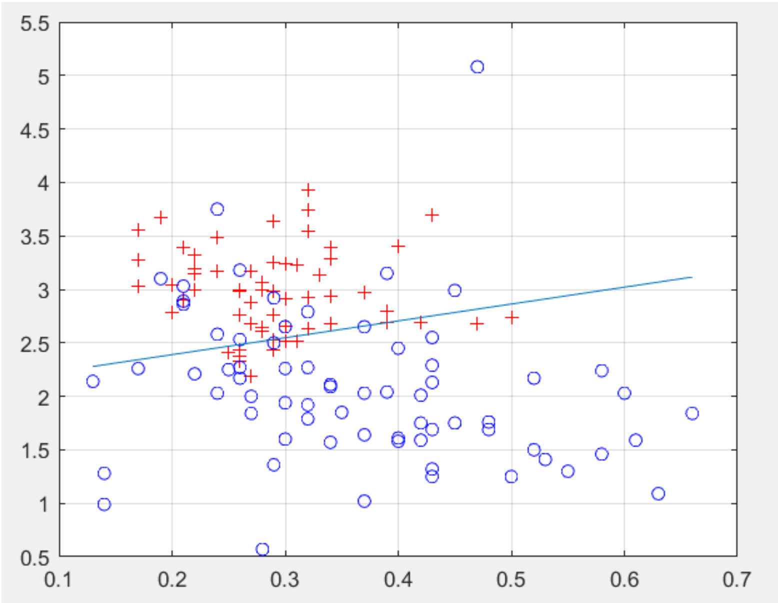
虽然结果并无太大区别，但是 logistic 回归的前提已经不再适用。
三个假设 1.条件独立 2.类条件分布高斯分布 3.方差和标签无关
结果应用到 UCI 数据集上

鸢尾花数据



结果很好

红酒数据



五、结论

1. logistic 回归对二分类问题具有很好的应用性
2. logistic 回归应用的条件是朴素贝叶斯前提：
1.)条件独立 2.)类条件分布高斯分布 3.)方差和标签无关
否则不再适用

六、参考文献

Pattern Recognition and Machine Learning
《机器学习实战》
《机器学习》周志华

七、附录：源代码（带注释）

```
%% 本函数的功能是根据所给分类方程，产生一组带标签的数据点(二维)
%  row 表示需要的数据点个数;
%  range_x、range_y 分别是数据[横纵坐标取值范围],都是 1*2 矩阵.
%  分类方程形式: y = a*x^m + b
function data = generate_signedData(row,range_x,range_y,a,m,b)
data=zeros(row,3);
min_x=range_x(1);
max_x=range_x(2);
min_y=range_y(1);
max_y=range_y(2);
for i=1:1:row
    %rand('seed',0);
    %初始化高斯随机生成器为0，可重复性
    x = min_x+(max_x-min_x)*rand(1,1);
    y = min_y+(max_y-min_y)*rand(1,1);
    sign = zeros(1,1);
    if y>(a*x^m + b)
        sign = 1;
    else
        sign = 0;
    end
    data(i,:)=[x,y,sign];
end
figure; %在图片中显示所有数据点
for i=1:1:row
    x1 = data(i,1);
    x2 = data(i,2);
    sign = data(i,3);
    if sign == 1
        plot(x1,x2,'r+','markersize',6);
    elseif sign == 0
        plot(x1,x2,'bo','markersize',6);
    end
    hold on;
end;
hold off;
grid on;

function Y=sigmoid(inX)
Y=1./(1+exp(-inX));
```

```
X=generate_signedData(100,[-8,8],[-8,8],2,1,2);
X0=ones(100,1);
X1=X(:,1);
X2=X(:,2);
Y=X(:,3);
alpha=0.001;
lamda=exp(-18);
maxcycle=500;
weights=ones(3,1);
dataMatrix=[X0,X1,X2];
for k=1:maxcycle
    h=sigmoid(dataMatrix*weights);
    error=Y-h;

weights=weights+alpha*dataMatrix'*error+lamda*weights'*weights
;
end
a0=weights(1);
a1=weights(2);
a2=weights(3);
y=(-a0*X0-a1*X1)./a2;
hold on;
plot(X1,y);

%%鸢尾花
A=load('iris.txt');
A=A(1:100,:);
A=A(:,[1 2 5]);
figure; %ÔÚÍ¼Æ¬ÖÐÏÔÊ¾ËùÓÐÊý¾Ýµã
for i=1:1:100
    x1 = A(i,1);
    x2 = A(i,2);
    sign = A(i,3);
    if sign == 1
        plot(x1,x2,'r+','markersize',6);
    elseif sign == 0
        plot(x1,x2,'bo','markersize',6);
    end
    hold on;
end;
hold off;
grid on;
X0=ones(100,1);
X1=A(:,1);
```

```
X2=A(:,2);
Y=A(:,3);
alpha=0.001;
maxcycle=500;
weights=ones(3,1);
dataMatrix=[X0,X1,X2];
for k=1:maxcycle
    h=sigmoid(dataMatrix*weights);
    error=Y-h;
    weights=weights+alpha*dataMatrix'*error;
end
a0=weights(1);
a1=weights(2);
a2=weights(3);
y=(-a0*X0-a1*X1)./a2;
hold on;
plot(X1,y);


%%红酒
A=load('wine.txt');
A=A(1:130,:);
A=A(:,[1 2 13]);
figure; %ÔÚÍ¼Æ¬ÖÐÏÔÊ¾ËùÓÐÊý¾Ýµã
for i=1:1:130

    x1 = A(i,2);
    x2 = A(i,3);
    sign = A(i,1);
    if sign == 1
        plot(x1,x2,'r+','markersize',6);
    elseif sign == 2
        plot(x1,x2,'bo','markersize',6);
    end
    hold on;
end;
hold off;
grid on;
X0=ones(130,1);
X1=A(:,2);
X2=A(:,3);
Y=A(:,1);
Y=Y-X0;
alpha=0.0001;
```

```
maxcycle=5000;
weights=ones(3,1);
dataMatrix=[X0,X1,X2];
for k=1:maxcycle
    h=sigmoid(dataMatrix*weights);
    error=Y-h;
    weights=weights+alpha*dataMatrix'*error;
end
a0=weights(1);
a1=weights(2);
a2=weights(3);
y=(-a0*X0-a1*X1)./a2;
hold on;
plot(X1,y);
```