哈爾濱工業大學

Harbin Institute of Technology

机器学习实验报告

逻辑回归

课程名称：	机器学习
学院：	计算学部
专业：	计算机科学与技术
学号：	**1180300811**
姓名：	孙骁
指导老师：	刘扬
学期：	**2020** 秋季学期

2020 年 **10** 月 **6** 日

目录

一、 实验目的和要求

1.1 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.

1.2 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

1.3 实验验证

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.

二、 实验环境

1. Anaconda 4.8.4
2. Python 3.7.4
3. PyCharm 2019.1 (Professional Edition)
4. Windows 10 2004

三、 实验原理

3.1 线性分类问题

考虑二分类问题，对于给定 d 个属性描述的示例 $\boldsymbol{x} = (x_1; x_2; \cdots; x_d)$，其中 x_i 是 $\boldsymbol{x}$ 在第 i 个属性上的取值. 有 $\boldsymbol{x} \in X$，$\boldsymbol{y} \in Y = \{0, 1\}$，即构建映射 $f: X \to Y$，即学习一个通过属性得线性组合进行预测的函数，

$$f(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b \tag{1}$$

改写成向量形式为

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b \tag{2}$$

对于离散的二分类问题，我们需要针对示例 $\boldsymbol{x}$ 进行类别的预测，比较两种类别的概率大小，将概率大的类别作为该示例的预测，即计算

$$P(Y|X) = \frac{P(XY)}{P(X)} \tag{3}$$

使用贝叶斯公式展开，得到

$$P(Y|X) = \frac{P(X|Y)\,P(Y)}{P(X)} \tag{4}$$

因此可以对给定的示例进行类别预测.

3.2 朴素贝叶斯

由于我们的目标是计算 $P(Y|X)$，又由贝叶斯定理可知，

$$P(Y|X) = \frac{P(X,Y)}{P(X)} = \frac{P(X|Y)\cdot P(Y)}{\sum_k P(X=x_i|Y)\,P(Y)} \tag{5}$$

由于分母的 $P(X)$ 是常数，所以只需要计算分子即可，即

$$P(Y=c_k),\ k=0,1 \tag{6}$$

$$P(X=x|Y=y_k) = P\left(X=x^{(1)},\cdots,X^{(d)}=x^{(d)}|Y=y_k\right),\ k=0,1 \tag{7}$$

但是由于条件概率分布 $P(X=x|Y=c_k)$ 具有指数级数量的参数，其估计实际上是不可行的. 因此，朴素贝叶斯对条件概率分布做了条件独立性的假设，由于这是一个较强的假设，由是也叫作“朴素”.

$$P(X=x\,|Y=c_k) = P\left(X^{(1)}=x^{(1)},\cdots,X^{(d)}=x^{(d)}\,|Y=y_k\right) \tag{8}$$

$$= \prod_{j=1}^{n} P\left(X^{(j)}=x^{(j)}\,|Y=y_k\right) \tag{9}$$

朴素贝叶斯法学习的是生成数据的机制，因此是一种学习模型. 条件独立假设即用于分类的特征在类确定的条件下都是条件独立的，这个假设使得整个算法变得简单，但是有时也会牺牲一些准确率.

因此，二分类的朴素贝叶斯的基本公式为

$$P(Y=c_k\,|X=x) = \frac{P(Y=y_k)\prod_j P\left(X^{(j)}=x^{(j)}\,|Y=y_k\right)}{\sum_k P(Y=y_k)\prod_j P\left(X^{(j)}=x^{(j)}\,|Y=y_k\right)},\ k=0,1 \tag{10}$$

朴素贝叶斯分类器可以表示为

$$y = f(x) = \arg\max_{c_k} \frac{P(Y=y_k)\prod_j P\left(X^{(j)}=x^{(j)}\,|Y=y_k\right)}{\sum_k P(Y=y_k)\prod_j P\left(X^{(j)}=x^{(j)}\,|Y=y_k\right)}$$

注意到分母对所有的 c_k 都是相同的，所以

$$y = f(x) = \arg\max_{c_k} P(Y = c_k) \prod_j P\left(X^{(j)} = x^{(j)} | Y = y_k\right)$$

3.3 Logistic 回归问题

Logistic 回归的基本思想就是利用朴素贝叶斯的假设计算 $P(Y|X)$: 即利用 $P(Y)$，$P(X|Y)$ 以及各个维度之间计算条件独立的假设来计算 $P(Y|X)$. 对于二分类问题，我们求解 $P(Y|X)$ 可以采用如下方式推导，

$$P(Y = 0|X) = \frac{P(Y = 0) P(X|Y = 0)}{P(X)} \tag{11}$$

对式 (11) 的分母应用全概率公式展开，得到

$$P(Y = 0|X) = \frac{P(Y = 0) P(X|Y = 0)}{P(X|Y = 0) P(Y = 0) + P(X|Y = 1) P(Y = 1)} \tag{12}$$

对式 (12) 右侧的分子分母同除 $P(Y = 0) P(X|Y = 0)$，得到

$$P(Y = 0|X) = \frac{1}{1 + \frac{P(X|Y = 1) P(Y = 1)}{P(X|Y = 0) P(Y = 0)}} \tag{13}$$

$$= \frac{1}{1 + \exp\left\{\ln \frac{P(X|Y = 1) P(Y = 1)}{P(X|Y = 0) P(Y = 0)}\right\}} \tag{14}$$

由于我们做的是二分类问题，故 Y 的分布满足伯努利分布，记 $\pi = \hat{P}(Y = 1)$，所以得到

$$P(Y = 0|X) = \frac{1}{1 + \exp\left\{\ln\left(\frac{\pi}{1-\pi}\right) + \ln\left(\frac{P(X|Y = 1)}{P(X|Y = 0)}\right)\right\}} \tag{15}$$

此处我们利用朴素贝叶斯的假设，假设各个特征之间的分布条件独立，因此有

$$P(Y = 0|X) = \frac{1}{1 + \exp\left\{\ln\left(\frac{\pi}{1-\pi}\right) + \sum_i \left(\ln\left(\frac{P(X_i|Y = 1)}{P(X_i|Y = 0)}\right)\right)\right\}} \tag{16}$$

此外，我们也假设各个维度的条件概率满足高斯分布，有各个维度的高斯分布函数 $P(X_i|Y = y_k) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(\frac{-(x-\mu_{ik})^2}{2\sigma_i^2}\right)$，将其带入，于是得到

$$P(Y = 0|X) = \frac{1}{1 + \exp\left\{\ln\left(\frac{\pi}{1-\pi}\right) + \sum_i \left(\frac{\mu_{i1} - \mu_{i0}}{\sigma_i^2} X_i + \frac{\mu_{i0}^2 - \mu_{i1}^2}{2\sigma_i^2}\right)\right\}} \tag{17}$$

将其转化为向量形式，可以得到

$$P(Y|X = 0) = \frac{1}{1 + \exp(\boldsymbol{w}^T \boldsymbol{X})} \tag{18}$$

其中，$\boldsymbol{w}_0 = \sum_i^n \left(\frac{\mu_{i0}^2 - \mu_{i1}^2}{2\sigma_i^2} \right)$，$\boldsymbol{w}_i = \frac{\mu_{i1} - \mu_{i0}}{\sigma_i^2}$，$i > 0$，$\boldsymbol{X} = (1, X_1, X_2, \cdots, x_n)$，由概率归一化的性质，我们可以得到

$$P(Y = 1|X) = \frac{\exp\left(\boldsymbol{w}^T \boldsymbol{X}\right)}{1 + \exp\left(\boldsymbol{w}^T \boldsymbol{X}\right)} \tag{19}$$

对于给定的数据集，我们使用极大似然法对参数 $\boldsymbol{w}$ 进行估计，于是有

$$\boldsymbol{w} = \arg\max_w P\left(Y^n | X^n, \boldsymbol{w}\right) \tag{20}$$

对式子做对数变换，因此我们的目标是最小化式 (21)

$$\mathcal{L}(\boldsymbol{w}) = \sum_n \left(-Y^n \boldsymbol{w}^T X + \ln\left(1 + \exp\left(\boldsymbol{w}^T \boldsymbol{X}\right)\right)\right) \tag{21}$$

为了防止模型过拟合，我们加入超参数 λ，得到式 (22)

$$\mathcal{L}(\boldsymbol{w}) = \frac{\lambda}{2} \boldsymbol{w}^T \boldsymbol{w} + \sum_n \left(-Y^n \boldsymbol{w}^T X + \ln\left(1 + \exp\left(\boldsymbol{w}^T \boldsymbol{X}\right)\right)\right) \tag{22}$$

四、 算法实现

4.1 梯度下降法

梯度下降法是通过迭代求目标函数最小值的一种方法，从数学上的角度来看，梯度的方向是函数增长速度最快的方向，那么梯度的反方向就是函数减少最快的方向. 由于上述的两种误差函数为二次型，因此应用梯度下降求得的局部最优解即为全局最优解.

对式 (22) 中的 $\boldsymbol{w}$ 应用梯度下降，在满足迭代误差保持在一定范围后，求得的 $\boldsymbol{w}$ 即为多项式的系数向量. 算法伪代码如1所示.

由于实验一中使用的梯度下降算法较为依赖数据分布，因此实验二将梯度下降算法进行了封装，同时为了避免数据量过大导致的溢出，我们将式 (22) 进行了归一化处理，得到

$$\mathcal{L}(\boldsymbol{w}) = \frac{\lambda}{2n} \boldsymbol{w}^T \boldsymbol{w} + \sum_n \left(-Y^n \boldsymbol{w}^T X + \ln\left(1 + \exp\left(\boldsymbol{w}^T \boldsymbol{X}\right)\right)\right) \tag{23}$$

Algorithm 1 Gradient Descent

Input: X, Y, penalty_coefficient, learning_rate, deviation

Output: w_result

```
Initialize X, Y, penalty_coefficient, learning_rate, deviation, data_number, feature, feature_number
Calculate pre_loss
while True do
    Update w with new gradient
```

Calculate new loss
if newloss < deviation **then**
break
else
if new loss > old loss **then**
Make the learning_rate half
end if
Update loss and w
end if
end while
return w

4.2 牛顿迭代法

牛顿迭代法的 $\boldsymbol{w}$ 更新方式与梯度下降法较为类似，详细的更新方式如式 (24) 与式 (25) 所示.

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \left(\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{w} \partial \boldsymbol{w}^T}\right)^{-1} \frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} \tag{24}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{w} \partial \boldsymbol{w}^T} = \sum_{i=1}^{n} \left(X_i X_i^T \frac{\exp\left(\boldsymbol{w}^T X\right)}{1+\exp\left(\boldsymbol{w}^T X\right)} \frac{1}{1+\exp\left(\boldsymbol{w}^T X\right)} \right) \tag{25}$$

算法伪代码如2所示.

Algorithm 2 Newton Iteration Descent

Input: X, Y, penalty_coefficient, learning_rate, deviation

Output: w_result

Initialize X, Y, penalty_coefficient, learning_rate, deviation, data_number, feature, feature_number
make pre_loss
while True **do**
Calculate new gradient
if the norm of new gradient < deviation **then**
break
end if
Calculate post_w
Update pre_w and post_w
end while
return w

五、 实验步骤

5.1 生成满足特征条件独立的数据并测试

设正例与反例的数据比为 1:1，生成数据，训练集与测试集的比例为 7:3，一共生成 100 个数据，均为二维数据，设置梯度下降算法的惩罚项为 30，具体代码见附录. 实现4.1节中的求解分类超平面系数向量的梯度下降算法和4.2节中的求解分类超平面系数向量的共轭梯度算法.

5.2 生成不满足特征条件独立的数据并测试

令二维度之间不相互独立，即协方差矩阵不为对角阵，各维度的方差不仅与特征相关，也与维度相关. 设正例与反例的数据比为 1:1，生成数据，训练集与测试集的比例为 7:3，一共生成 100 个数据，均为二维数据，设置梯度下降算法的惩罚项为 30，具体代码见附录. 实现4.1节中的求解分类超平面系数向量的梯度下降算法和4.2节中的求解分类超平面系数向量的共轭梯度算法.

5.3 使用 UCI 数据集进行 Logistic 分类

选用 UCI 钞票数据集，转弯时一个根据钞票的图像判断钞票真伪的数据集，对图像进行小波变换后提取相应的特征. 数据集中一共有 1372 个样本，每条数据包括一个分类标签和四个特征，四个特征分别是：

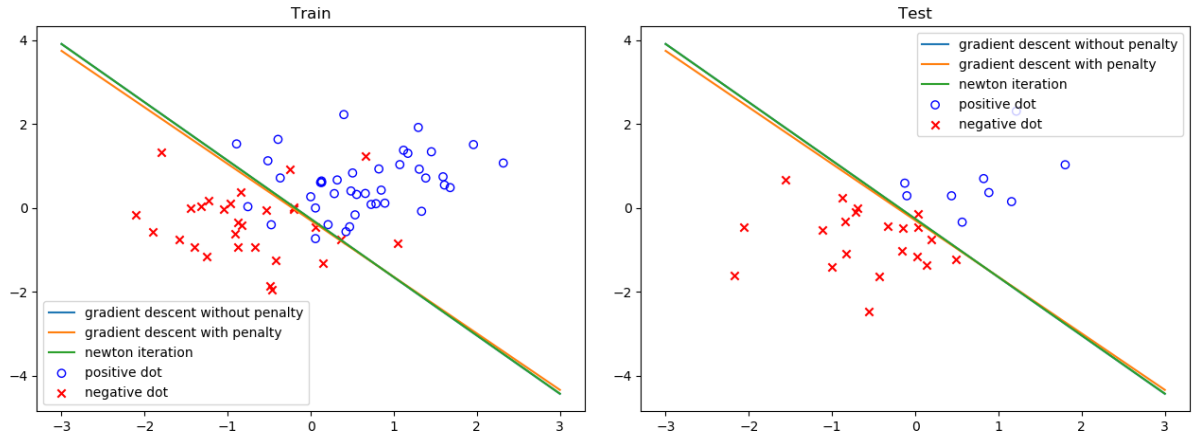
1. 原始图像经过小波变换后的方差 (variance)
2. 原始图像经过小波变换后的偏态 (skewness)
3. 原始图像经过小波变换后的峰度 (curtosis)
4. 原始图像的熵 (entropy)

数据集没有数据特征缺失的情况，故按照训练集与测试集 7:3 的比例划分，选择惩罚项为 10. 实现4.1节中的求解分类超平面系数向量的梯度下降算法和4.2节中的求解分类超平面系数向量的共轭梯度算法. 因为是数据特征是四维，无法画出特征空间的分布，故只显示测试的准确率.

六、 实验结果

6.1 特征相互独立时的分类结果

实验结果如图 (1) 所示. 三种方法在训练集上的分类准确率均为 96.67%，在测试集上的分类准确率为 8.57%.

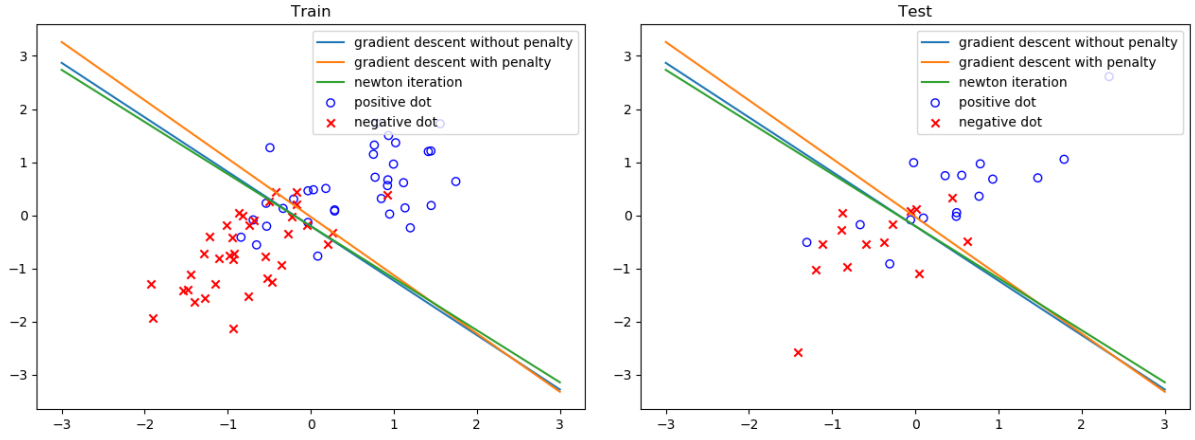


(a) 特征间相互独立时训练集分类结果　　(b) 特征间相互独立时测试集分类结果

图 **1**　特征相互独立时的分类结果

6.2 特征不满足相互独立时的分类结果

实验结果如图 (2) 所示. 不带惩罚项的梯度下降法与带惩罚项的梯度下降法在训练集上的分类准确率均为 82.86%，牛顿迭代法在训练集上的的分类准确率为 84.29%；不带有惩罚项的梯度下降法和牛顿迭代法在测试集上的分类准确率为 76.67%，带有惩罚项的梯度下降法在测试集上的分类准确率为 73.33%.



(a) 特征间不相互独立时训练集分类结果　　(b) 特征间不相互独立时测试集分类结果

图 **2**　特征相互独立时的分类结果

6.3 在 UCI 数据集上测试结果

由于 UCI 数据集 banknote authentication Data Set 为四维数据，所以无法画出数据在特征空间中的分布，但是得到如图 (3) 所示，在训练集和测试集上的分类准确率. 在训练集上，不带有惩罚项和带有惩罚项的梯度下降法的分类准确率均为 94.27%，牛顿迭

```
----------start, the bank note authentication data set-------------
The accuracy on train data set:
Gradient descent without penalty's accuracy:  0.9427083333333334
Gradient descent with penalty's accuracy:  0.9427083333333334
Newton iteration:  0.94375

The accuracy on test data set:
Gradient descent without penalty's accuracy:  0.9951456310679612
Gradient descent with penalty's accuracy:  0.9878640776699029
Newton iteration:  0.9951456310679612
```

图 **3**　在 **UCI** 数据集上测试结果

代法的分类准确率是 94.38%；在测试集上，不带有惩罚项的梯度下降法和牛顿迭代法的分类准确率均为 99.51%，带有惩罚项的梯度下降法的分类准确率为 98.79%.

七、 实验结论

1. 数据量较大时，惩罚项对分类超平面的影响逐渐降低；
2. 牛顿迭代法与梯度下降法均可以得到较好的实验结果，分类结果较为准确；
3. 在数据特征较少时，尽管不满足特征间相互独立的条件，Logistic 回归仍然可以做到较好的分类，当数据特征较多时，如果不满足特征间相互独立的条件，分类结果较差；
4. Logistics 回归可以很好地解决简单的线性分类问题，而且收敛速度较快.

参考文献

[1] 李航, 统计学习方法 (2019.3).

[2] 周志华, 机器学习 (2016.1).

[3] Banknote Authentication Data Set. (2012.8) [Data set]