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：必修

实验题目：logistics Regression

学号：1160300314

姓名：朱明彦

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的网站上，找一实际数据加以测试。

实验环境

- **OS**: Ubuntu 16.04.5 LTS
- python 3.7.0

三、设计思想(本程序中用到的主要算法及数据结构)

1.算法原理

Logistic回归的**基本思想**就是利用朴素贝叶斯的假设取计算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$(Y|X)$。

考虑二分类问题，$f: X \rightarrow Y$，其中X为实数向量, $X =< X_1, X_2, \dots, X_n >$，$Y \in \{0, 1\}$，且有对于所有的$X_i$在给定$Y$的前提下均有条件独立(即各维条件独立)成立；并且有P$(X_i|Y=y_k) \sim N(\mu_{ik}, \sigma_i)$，P$(Y) \sim B(\pi)$成立。

那么我们求解P$(Y|X)$的方式，就可以有下面这种方式的推导：

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

将式(1)下面全概率公式展开可以得到：

$$\mathsf{P}(Y=0|X)=\frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)+P(Y=0)P(X|Y=0)} \tag{2}$$

将(2)式右边，上下同时除以$P(Y=0)P(X|Y=0)$，得到：

$$P(Y=0|X)=\frac{1}{1+\frac{P(Y=1)P(X|Y=1)}{P(Y=0)P(X|Y=0)}}=\frac{1}{1+exp\left(ln\frac{P(Y=1)P(X|Y=1)}{P(Y=0)P(X|Y=0)}\right)} \tag{3}$$

又由于Y符合伯努利分布，我们可以将$\pi=\widehat{P}(Y=1)$带入(3)中得到:

$$P(Y=0|X)=\frac{1}{1+exp\left(ln(\frac{\pi}{1-\pi})+ln\frac{P(X|Y=1)}{P(X|Y=0)}\right)} \tag{4}$$

因为有着朴素贝叶斯的假设，我们可以讲向量各维的分布展开：

$$P(Y=0|X)=\frac{1}{1+exp\left(ln(\frac{\pi}{1-\pi})+\sum_i\left(ln\frac{P(X_i|Y=1)}{P(X_i|Y=0)}\right)\right)} \tag{5}$$

另由于各个维度使符合高斯分布的，我们可以将各个维度的高斯分布函数$P(X_i|Y=y_k)=\frac{1}{\sigma_{ik}\sqrt{2\pi}}exp(\frac{-(x-\sigma_{ik})^2}{2\sigma_{ik}^2})$带入，可以有:

$$P(Y=0|X)=\frac{1}{1+exp\left(ln(\frac{\pi}{1-\pi})+\sum_i\left(\frac{\mu_{i1}-\mu_{i0}}{\sigma_i^2}X_i+\frac{\mu_{i0}^2-\mu_{i1}^2}{2\sigma_i^2}\right)\right)} \tag{6}$$

将其写为向量的形式，可以转化为：

$$P(Y=0|X)=\frac{1}{1+exp(\mathbf{w^TX})} \tag{7}$$

其中，$\mathbf{w}_0=\sum_i^n\left(\frac{\mu_{i0}^2-\mu_{i1}^2}{2\sigma_i^2}\right)+ln(\frac{\pi}{1-\pi})$，$\mathbf{w}_i=\frac{\mu_{i1}-\mu_{i0}}{\sigma_i^2},\ i>0,\mathbf{X}=\begin{bmatrix}1\\X_1\\\vdots\\X_n\end{bmatrix}$

利用归一化的特性，我们可以得到:

$$P(Y=1|X)=1-\frac{1}{1+exp(\mathbf{w^TX})}=\frac{exp(\mathbf{w^TX})}{1+exp(\mathbf{w^TX})} \tag{8}$$

当前我们有(或者有生成的数据) $\{<X^1,Y^1>,\dots,<X^l,Y^l>\}$，我们通过最大条件似然法对参数$\mathbf{w}$进行估计：

$$\mathbf{w}_{MCLE} = arg\max_{\mathbf{w}} \prod_l P(Y^l|X^l, \mathbf{w}) \tag{9}$$

对式(9)两边同时取对数，我们可以有：

$$l(\mathbf{w}) \equiv ln\prod_l P(Y^l|X^l, \mathbf{w}) = \sum_l P(Y^l|X^l, \mathbf{w}) \tag{10}$$

将式$(8)(9)$带入(10)中，我们得到：

$$l(\mathbf{w}) = \sum_l \left(Y^l \mathbf{w}^{\mathbf{T}}\mathbf{X} - ln(1 + exp(\mathbf{w}^{\mathbf{T}}\mathbf{X}))\right) \tag{11}$$

最大化式(11)就等价于最小化式(12):

$$\mathcal{L}(\mathbf{w}) = \sum_l \left(-Y^l \mathbf{w}^{\mathbf{T}}\mathbf{X} + ln(1 + exp(\mathbf{w}^{\mathbf{T}}\mathbf{X}))\right) \tag{12}$$

式(12)是关于$\mathbf{w}$的高阶可导连续凸函数[2]，根据凸优化的理论，在这里我们可以用梯度下降法、牛顿法等求解其最优解，在算法实现方面详述。

为了避免过拟合现象，我们仿照lab1的经验，对于式(12)增加惩罚项，其中λ为超参数：

$$\mathcal{L}(\mathbf{w}) = \frac{\lambda}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + \sum_l \left(-Y^l \mathbf{w}^{\mathbf{T}}\mathbf{X} + ln(1 + exp(\mathbf{w}^{\mathbf{T}}\mathbf{X}))\right) \tag{13}$$

2.算法的实现

算法实现部分，此处选择使用梯度下降法实现以及使用牛顿法进行优化。

2.1 梯度下降实现

根据Lab1的经验，其实对于梯度下降法的使用没有什么变化，只是将优化的函数做了一下修改，所以我们可以得到其一阶导数，然后在$t+1$轮得到的迭代式子如下，其中α为学习率:

$$\mathbf{w}^{\mathbf{t+1}} = \mathbf{w}^{\mathbf{t}} - \alpha\frac{\partial \mathcal{L}}{\partial \mathbf{w}}(\mathbf{w}^{\mathbf{t}})$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = -\sum_{i=1}^{l} X_i \left(Y_i - \frac{exp(\mathbf{w}^{\mathbf{T}}\mathbf{X})}{1 + exp(\mathbf{w}^{\mathbf{T}}\mathbf{X})}\right)$$

这种直接将式(13)求导进行迭代的方式，存在在数据特别多(即l特别大)的情况下，有可能导致上溢出发生，基于此，我们将式(13)归一化，防止其溢出，得到式(14):

$$\mathcal{L}(\mathbf{w}) = \frac{\lambda}{2l}\mathbf{w}^{\mathbf{T}}\mathbf{w} + \frac{1}{l}\sum_l \left(-Y^l \mathbf{w}^{\mathbf{T}}\mathbf{X} + ln(1 + exp(\mathbf{w}^{\mathbf{T}}\mathbf{X}))\right) \tag{14}$$

然后再进行迭代，就可以避免上溢出的现象。

2.2 牛顿法实现

与梯度下降法实现类似，此处我们有在$t+1$轮迭代的式子如下:

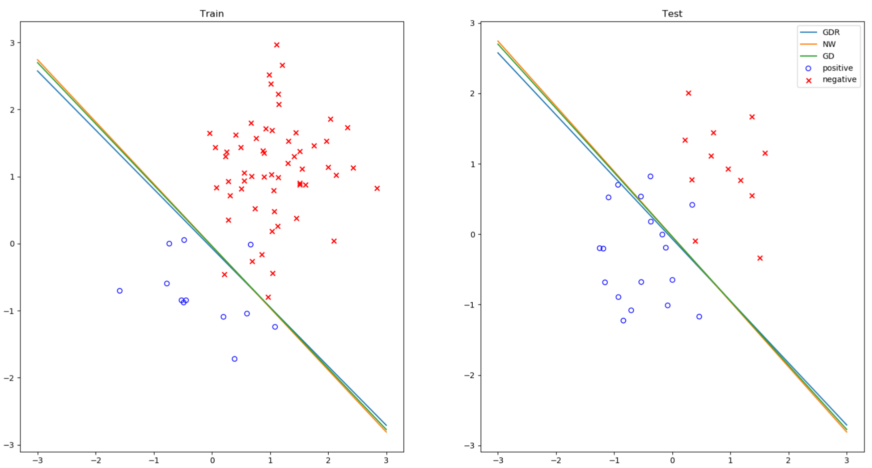
$$\mathbf{w^{t+1}} = \mathbf{w^t} - \left(\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w} \partial \mathbf{w^T}}\right)^{-1} \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$$

$$\frac{\partial^2 a \mathcal{L}}{\partial \mathbf{w} \partial \mathbf{w^T}} = \sum_{i=1}^{l} \left(X_i X_i^T \frac{exp(\mathbf{w^T X})}{1 + exp(\mathbf{w^T X})} \frac{1}{1 + exp(\mathbf{w^T X})} \right)$$

四、实验结果分析

1.利用生成数据，符合所有假设

设置正反例比例为3:7，生成数据，训练集与测试集的比例为3:7，共生成100个测试用例，均为二维数据，超参数$\lambda = 0.1$得到的测试结果如下：



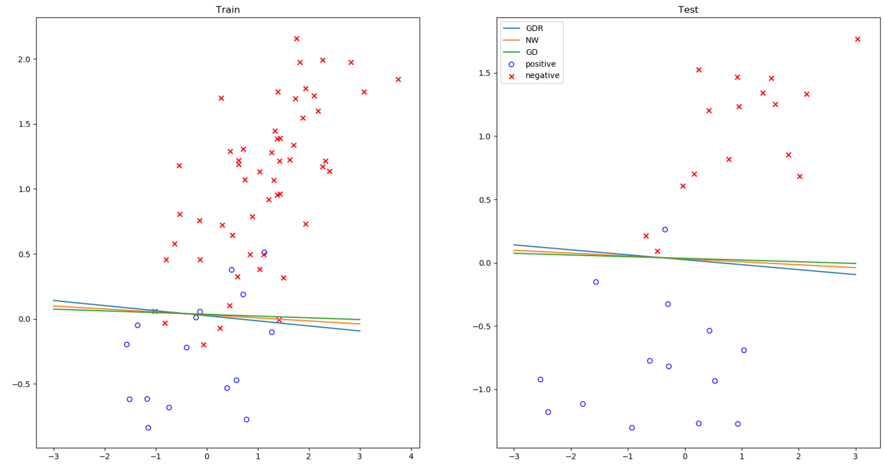
左侧为训练集，右侧为测试集，三种方式的准确率均为0.9.

2.破坏各个维度之间的条件独立性

即，将协方差矩阵进行设置，使其不为对角阵，具体

```
generating_x, generating_y = generate_2_dimension_data(number_gen,
mean_gen_pos, mean_gen_neg, proportion_pos_gen,cov21=1)
```

其余条件保持不变，得到的测试结果如下



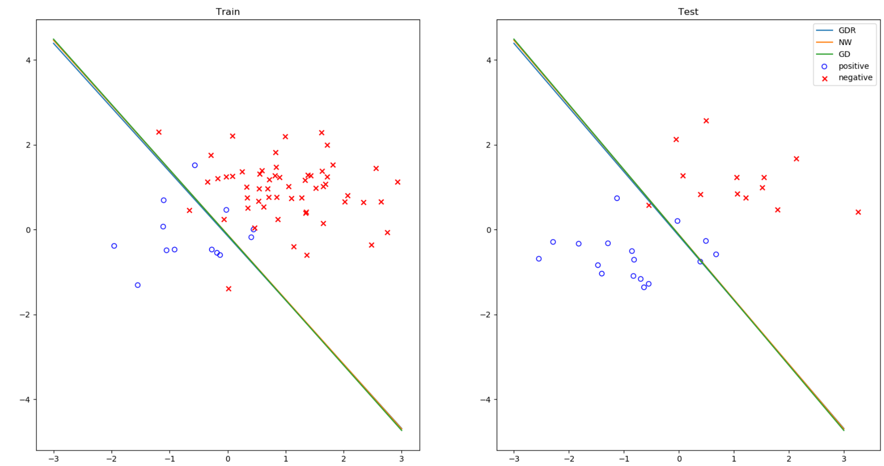
牛顿法、梯度下降(带或不带正则项)准确率均为0.967

3.破坏方差仅与类别相关，而与维度无关的条件

即，将各个维度的方差增加不同的偏置，以达到使各个维度的方差不仅与类别相关，还与维度相关。 具体的操作在lab2中我们使用下面的方式

```
generating_x, generating_y = generate_2_dimension_data(number_gen, me
mean_gen_neg, proportion_pos_gen,scale_pos1_bios=0.3,scale_neg1_bios=
```

其余条件保持不变，得到的结果如下：



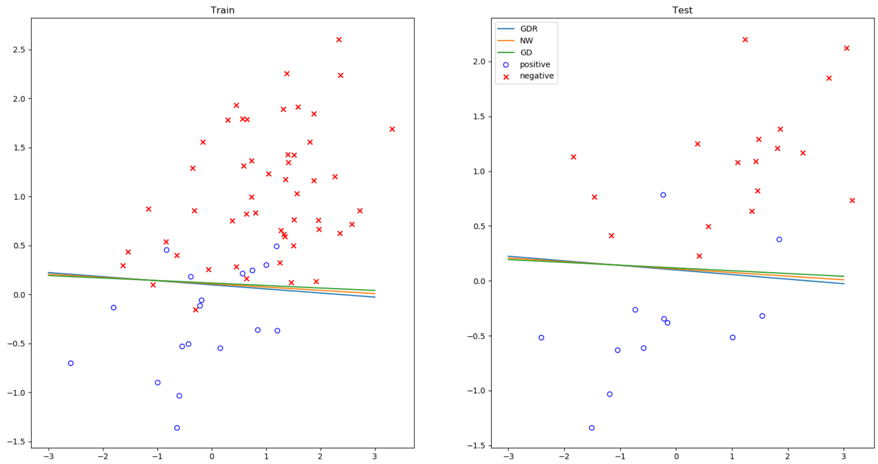
准确率均为0.867

4.同时破坏2.3.两个条件

使用下面的语句

```
generating_x, generating_y = generate_2_dimension_data(
    number_gen,mean_gen_pos, mean_gen_neg, proportion_pos_gen,cov21=
    scale_pos1_bios=0.3,scale_neg1_bios=0.6)
```

测试的结果如下：



准确率均为0.933

经过上面4种对比实验，分别对于条件满足和不满足的3种情况，我们可以看到对于两种方法得到的最优解是十分接近的，并且准确率也十分接近(以上仅列出较少的测试结果)，在实验时经过多次测试得到的结果，可以看出梯度下降法(带不带正则项)与牛顿法的准确率都十分接近。

5.使用UCI上数据进行测试

使用的数据为钞票数据集Banknote Dataset

这是从纸币鉴别过程中的图像里提取的数据，用来预测钞票的真伪的数据集。该数据集中含有1372个样本，每个样本由5个数值型变量构成，4个输入变量和1个输出变量。小波变换工具用于从图像中提取特征。这是一个二元分类问题。

每一行的5个(列)变量含义如下：

第一列：图像经小波变换后的方差(variance)(连续值)；

第二列：图像经小波变换后的偏态(skewness)(连续值)；

第三列：图像经小波变换后的峰度(curtosis)(连续值)；

第四列：图像的熵(entropy)(连续值)；

第五列：钞票所属的类别a(整数，0或1)。

首先将文件从`data_banknote_authentication.csv`中读出(使用`bank_note_read.py`)，并对数据是否有缺失进行统计，统计的结果如下：

```
   variance  skewness  kurtosis  entropy  a
0   3.62160    8.6661   -2.8073 -0.44699  0
1   4.54590    8.1674   -2.4586 -1.46210  0
2   3.86600   -2.6383    1.9242  0.10645  0
3   3.45660    9.5228   -4.0112 -3.59440  0
4   0.32924   -4.4552    4.5718 -0.98880  0

variance    0
skewness    0
kurtosis    0
entropy     0
a           0
```

其中上半部分为整个数据的前5行数据，后半部分为对于数据中为`null`的数据进行统计，可以看到对于5列属性来说，**均没有缺失数据的现象**，故可以直接进行训练。

此处选择的训练集与测试集的比例为3:7，超参数λ选择0.1

测试结果如下：

测试	GD Accuracy	NW Accuracy
1	0.995	0.995
2	0.998	0.998
3	0.985	0.985
4	0.99	0.99
5	0.987	0.987

五、结论

- 对于梯度下降法，是否使用惩罚项对于测试结果的影响不大。
- 牛顿法与梯度下降法都可以得到很好的结果，其中梯度下降法的迭代收敛的速度较慢。
- 对于进行数学推导时使用的假设(包括朴素贝叶斯假设和σ仅与类别相关与维度无关)，当打破假设时，得到的分类结果仍然较好，这与[Domingos&Pazzani, 1996]中的阐述相符合。

六、参考文献

- Christopher Bishop. Pattern Recognition and Machine Learning.

- 周志华 著. 机器学习, 北京: 清华大学出版社, 2016.1
- Newton's Method, wiki
- Blood Transfusion Service Center Data Set. UCI
- Banknote Dataset. UCI
- Mushroom Dataset. UCI
- Domingos, P. & M. Pazzani (1996). Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In L. Saitta (Ed.), Proceedings of the Thirteenth International Conference on Machine Learning (pp. 105–112). San Francisco, CA: Morgan Kaufmann.

七、附录:源代码(带注释)

- 此处不再单独给出，主程序见`lab2.py`
- 梯度下降程序见`gradient_descent.py`
- 牛顿法程序见`newton_method.py`
- 读取UCI钞票数据集见`bank_note_read.py`
- 读取UCI献血数据见`blood_read.py`，读取UCI蘑菇数据见`mushroom_read.py`。此处单独进行有关UCI中蘑菇是否有毒的测试以及是否进行献血的测试，原因在于其各个属性均以离散的形式给出，想使用上述方法进行处理，必然涉及到数据预处理相关问题，这种数据的形式与我们之前所使用高斯分布推出的结论不相适合，故此处没有再单独进行阐述，**总的来说，虽然在连续的属性上推出上述结论，但是在离散的比那里上work仍然很好，正确率同样很高**。