# 机器学习导论
# 作业二

151220129, 吴政亿, 18805156360@163.com

2018 年 5 月 8 日

## 1 [25pts] Multi-Class Logistic Regression

教材的章节3.3介绍了对数几率回归解决二分类问题的具体做法。假定现在的任务不再是二分类问题，而是多分类问题，其中标记$y \in \{1, 2 \ldots, K\}$。请将对数几率回归算法拓展到该多分类问题。

(1) **[15pts]** 给出该对率回归模型的“对数似然”(log-likelihood);

(2) **[10pts]** 计算出该“对数似然”的梯度。

提示1：假设该多分类问题满足如下$K-1$个对数几率，

$$\begin{aligned} \ln \frac{p(y=1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_1^{\mathrm{T}}\mathbf{x} + b_1 \\ \ln \frac{p(y=2|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_2^{\mathrm{T}}\mathbf{x} + b_2 \\ &\cdots \\ \ln \frac{p(y=K-1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x} + b_{K-1} \end{aligned}$$

提示2：定义指示函数$\mathbb{I}(\cdot)$，

$$\mathbb{I}(y=j) = \begin{cases} 1 & \text{若}y\text{等于}j \\ 0 & \text{若}y\text{不等于}j \end{cases}$$

**Solution.** *此处用于写解答(中英文均可)*

*(1)* 由提示*1*得

$$\begin{aligned} \frac{p(y=i|\mathbf{x})}{p(y=K|\mathbf{x})} &= e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i} \\ \frac{\sum_{i=1}^{K-1} p(y=i|\mathbf{x})}{p(y=K|\mathbf{x})} = \frac{1-p(y=K|\mathbf{x})}{p(y=K|\mathbf{x})} &= \sum_{i=1}^{K-1} e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i} \end{aligned}$$

得到

$$\begin{aligned} p(y=K|\mathbf{x}) &= \frac{1}{1+\sum_{i=1}^{K-1} e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i}} \\ p(y=n|\mathbf{x}) &= \frac{e^{\mathbf{w}_n^{\mathrm{T}}\mathbf{x}+b_n}}{1+\sum_{i=1}^{K-1} e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i}} \end{aligned}$$

同书上$p59$的简写,对数似然为

$$\begin{aligned} \ell(\beta) &= \sum_{n=1}^{m} \ln p(y=n|\mathbf{x}) \\ &= \sum_{n=1}^{m} \ln \prod_{i=1}^{K} p_i(\mathbf{x}_n)^{\mathbb{I}(y_n=i)} \\ &= \sum_{n=1}^{m} \sum_{i=1}^{K} \mathbb{I}(y_n=i) \ln p_i(\mathbf{x}_n) \\ &= \sum_{n=1}^{m} [\sum_{i=1}^{K-1} \mathbb{I}(y_n=i)\beta_i^{\mathrm{T}}\hat{\mathbf{x}_n} - \ln(1+\sum_{j=1}^{K-1} e^{\beta_j^{\mathrm{T}}\hat{\mathbf{x}_n}})] \end{aligned}$$

*(2)* 对每一个$\beta$求偏导，得到该对数似然的梯度为*:*

$$\frac{\partial \ell(\beta)}{\partial \beta_t} = \sum_{n=1}^{m} \hat{\mathbf{x}_n}[\mathbb{I}(y_n=t) - \frac{e^{\beta_t^{\mathrm{T}}\hat{\mathbf{x}_n}}}{1+\sum_{i=1}^{K-1} e^{\beta_i^{\mathrm{T}}\hat{\mathbf{x}_n}}}]$$

## 2 [20pts] Linear Discriminant Analysis

假设有两类数据，正例独立同分布地从高斯分布$\mathcal{N}(\mu_1, \Sigma_1)$采样得到，负例独立同分布地从另一高斯分布$\mathcal{N}(\mu_2, \Sigma_2)$采样得到，其中参数$\mu_1, \Sigma_1$及$\mu_2, \Sigma_2$均已知。现在，我们定义“最优分类”：若对空间中的任意样本点，分别计算已知该样本采样于正例时该样本出现的概率与已知该样本采样于负例时该样本出现的概率后，取概率较大的所采类别作为最终预测的类别输出，则我们说这样的分类方式满足“最优分类”性质。

试证明：当两类数据的分布参数$\Sigma_1 = \Sigma_2 = \Sigma$时，线性判别分析 (LDA)方法满足“最优分类”性质。（提示：找到满足最优分类性质的分类平面。）

**Solution.** 此处用于写解答*(*中英文均可*)*

应用《机器学习》书中第七章的贝叶斯分类器，得到贝叶斯分类器为

$$h^*(x) = \arg\max_{c \in \mathcal{Y}} P(c|x)$$

在*LDA*中，已知$\Sigma_1 = \Sigma_2 = \Sigma$下，已知多变量的高斯分布密度函数为*:*

$$f_c(x) = \frac{1}{(2\pi)^{d/2}|\Sigma_c|^{1/2}} e^{-\frac{1}{2}(x-\mu_c)^T \Sigma_c^{-1}(x-\mu_c)}$$

得到

$$\begin{aligned} h^*(x) &= \arg\max_{c \in \mathcal{Y}} P(h = c|X = x) \\ &= \arg\max_{c \in \mathcal{Y}} f_c(x) P(c) \\ &= \arg\max_{c \in \mathcal{Y}} \log(f_c(x) P(c)) \\ &= \arg\max_{c \in \mathcal{Y}} \left[ -\frac{1}{2}(x-\mu_c)^T \Sigma^{-1}(x-\mu_c) + \log(P(c)) \right] \end{aligned}$$

因此

$$h^*(x) = \arg\max_{c \in \mathcal{Y}} \left[ x^T \Sigma^{-1} \mu_c - \frac{1}{2} \mu_c^T \Sigma^{-1} \mu_c + \log(P(c)) \right]$$

当数据*1*与数据*2*代入$h^*(x)$相同时，即为满足最优分类性质的分类平面，即

$$\log \frac{P(m_1)}{P(m_2)} - \frac{1}{2}(\mu_{m_1} + \mu_{m_2})^T \Sigma^{-1}(\mu_{m_1} - \mu_{m_2}) + x^T \Sigma^{-1}(\mu_{m_1} - \mu_{m_2}) = 0$$

由于此函数为线性函数，故为满足最优分类性质的分类平面，平面法向量为

$$\omega^* = \Sigma^{-1}(\mu_{m_1} - \mu_{m_2})$$

[1]

[1]参考:https://blog.csdn.net/VictoriaW/article/details/78275394?locationNum=8&fps=1

# 3 [55+10*pts] Logistic Regression Programming

在本题中，我们将初步接触机器学习编程，首先我们需要初步了解机器学习编程的主要步骤，然后结合对数几率回归，在UCI数据集上进行实战。机器学习编程的主要步骤可参见博客。

本次实验选取UCI数据集Page Blocks（下载链接）。数据集基本信息如表 1所示，此数据集特征维度为10维，共有5类样本，并且类别间样本数量不平衡。

表 1: Page Blocks数据集中每个类别的样本数量。

| 标记 | 1 | 2 | 3 | 4 | 5 | total |
|---|---|---|---|---|---|---|
| 训练集 | 4431 | 292 | 25 | 84 | 103 | 4935 |
| 测试集 | 482 | 37 | 3 | 4 | 12 | 538 |

对数几率回归（Logistic Regression, LR）是一种常用的分类算法。面对多分类问题，结合处理多分类问题技术，利用常规的LR算法便能解决这类问题。

(1) [**5pts**] 此次编程作业要求使用Python 3或者MATLAB编写，请将main函数所在文件命名为 LR_main.py或者LR_main.m，效果为运行此文件便能完成整个训练过程，并输出测试结果，方便作业批改时直接调用；

(2) [**30pts**] 本题要求编程实现如下实验功能：

- [**10pts**] 根据《机器学习》3.3节，实现LR算法，优化算法可选择梯度下降，亦可选择牛顿法；
- [**10pts**] 根据《机器学习》3.5节，利用“一对其余”（One vs. Rest, OvR）策略对分类LR算法进行改进，处理此多分类任务；
- [**10pts**] 根据《机器学习》3.6节，在训练之前，请使用“过采样”（oversampling）策略进行样本类别平衡；

(3) [**20pts**] 实验报告中报告算法的实现过程（能够清晰地体现 (1) 中实验要求，请勿张贴源码），如优化算法选择、相关超参数设置等，并填写表 **??**，在`http://www.tablesgenerator.com/`上能够方便地制作LaTex表格；

(4) [**附加题 10pts**] 尝试其他类别不平衡问题处理策略（尝试方法可以来自《机器学习》也可来自其他参考材料），尽可能提高对少数样本的分类准确率，并在实验报告中给出实验设置、比较结果及参考文献；

[**注意**] 本次实验除了numpy等数值处理工具包外禁止调用任何开源机器学习工具包，一经发现此实验题分数为0，请将实验所需所有源码文件与作业pdf文件放在同一个目录下，请勿将数据集放在提交目录中。

**实验报告.**

(1) 我已遵循实验要求，应用python3编程，文件名为**LR_main.py**，可直接运行并输出各标记查准率与查全率，以及准确率。

(2) (a) 实现LR算法，并且分别应用了**gradAscent**与**newTon**，可以在main函数中更改bool变量**isNewton = True**进行切换

(b) 分别将样本1 5作为正类，其他作为反类，应用LR算法，本代码对应函数
**def classify(num, isOverSampling, isNewton)**
其中num为对应的作为正类的label，isOverSampling选择是否过采样，isNewton选择优化算法

(c) 过采样应用了抽样采样，设置一个随机数令结果更具有随机性，在loadTrainData时调用并进行归一化(线性归一化)，对应函数**def overSampling(feature, label)**

(3) 牛顿迭代:每次按照教材公式求一阶导数与二阶导数，然后对比迭代前后的向量欧氏距离，当欧氏距离小于1e-1时停止迭代，实验中遇到了numpy.exp溢出问题，最后将公式变形为$\frac{1}{1+e^{w}x+b}$后得到改善，但扔有溢出现象发生，主要在标记二中梯度下降：每次步长固定为0.01，对于迭代次数，大量具有重复性的精准良好结果均证明500达到局部最优值

表 2: 无过采样梯度下降步长0.01 迭代500

| 标记 | 1 | 2 | 3 | 4 | 5 | 准确率 |
|---|---|---|---|---|---|---|
| 查全率 | 0.98 | 0.81 | 0.67 | 0.75 | 0.25 | 0.95 |
| 查准率 | 0.97 | 0.77 | 1.00 | 1.00 | 0.50 | |

表 3: 过采样(随机) 梯度下降步长0.01 迭代500

| 标记 | 1 | 2 | 3 | 4 | 5 | 准确率 |
|---|---|---|---|---|---|---|
| 查全率 | 0.93 | 0.78 | 1.00 | 1.00 | 0.33 | 0.95 |
| 查准率 | 0.97 | 0.97 | 0.17 | 0.40 | 0.21 | |

表 4: 过采样(随机) 牛顿迭代

| 标记 | 1 | 2 | 3 | 4 | 5 | 准确率 |
|---|---|---|---|---|---|---|
| 查全率 | 0.91 | 0.95 | 0.00 | 0.00 | 0.75 | 0.90 |
| 查准率 | 0.99 | 0.73 | 0.00 | 0.00 | 0.20 | |

由表中数据可以得到，过采样对标记2345等样本较少的标记是友好的，但是对标记1不太友好，另外实验数据中无过采样表现出更高的准确率，对于过采样其中的误差可能因为忽略了某些重要的点，或者某些不具有实际意义的点在过采样中被随机出现的次数过高，因此过采样的数据具有较大波动性，上面表格为程序运行后选择的最具有代表性的值。

下面表格5简要描述核心算法

表 5: 核心算法介绍

| 函数 | 含义 |
|---|---|
| firstDerivative | 对给定的书中的β求一阶导数 |
| secondDerivative | 对给定的书中的β求二阶导数 |
| newton | 牛顿迭代，每一次的差值为- secondDerivativeˆ-1 * firstDerivative |
| gradAscent | 梯度下降，每一次的差值为步长*featureˆ-1*(label - firstDerivative) |
| overSampling | 过采样，得到最大的标记数，然后将较少的标记类通过随机取样的方式扩充 |
| classify | 将训练集的num作为正类，其他作为反类，计算并返回权值 |

表 6: 再缩放梯度下降步长0.01 迭代500

| 标记 | 1 | 2 | 3 | 4 | 5 | 准确率 |
|---|---|---|---|---|---|---|
| 查全率 | 0.95 | 0.89 | 1.00 | 1.00 | 0.67 | 0.95 |
| 查准率 | 0.99 | 0.67 | 0.67 | 0.50 | 0.17 | |

表 7: 再缩放牛顿迭代

| 标记 | 1 | 2 | 3 | 4 | 5 | 准确率 |
|---|---|---|---|---|---|---|
| 查全率 | 0.98 | 0.89 | 1.00 | 0.33 | 0.75 | 0.95 |
| 查准率 | 0.98 | 0.83 | 0.67 | 1.00 | 0.25 | |

表 8: 过采样(smote) 牛顿迭代

| 标记 | 1 | 2 | 3 | 4 | 5 | 准确率 |
|---|---|---|---|---|---|---|
| 查全率 | 0.09 | 0.51 | 0.00 | 0.00 | 0.08 | 0.12 |
| 查准率 | 0.95 | 0.10 | 0.00 | 0.00 | 0.01 | |

(4) 应用**再放缩**得到如下数据，其中梯度下降的步长为0.01，迭代次数为500;牛顿法在欧氏距离小于0.1时停止

可以发现，对于梯度下降，再缩放对于查全率有着明显上升，但是查准率有所下降，由此分析该模型可能存在一定程度的欠拟合；同理，对于牛顿迭代，再缩放对于查全率有所下降，但是在查准率明显上升，由此分析该模型可能存在一定程度的过拟合。

另外，对于smote算法，将代码95-125注释，将code 67-92取消注释即可进行smote，从理论上来讲，我们可以得到smote算法对于少数的标记具有更好的拟合，然而事实上实验结果恰恰相反，这其中的误差可能是因为我们生成了许多无效点。在这里我提出一种判断生成的smote点是否时有效点的方法：

我们可以通过计算生成的smote点在一定范围内，正例的点个数与反例的点个数的数量(正例指的是smote生成的点类型)，如果反例的点与正例的点的比值大于一个定值，那我们认为这个点无效，重新生成。

[2]

[2]参考:https://www.cnblogs.com/muchen/p/6296957.html