机器学习导论
作业二

151220097, 孙旭东, 248381185@qq.com

2018 年 4 月 14 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. *1.* 因为

$$\ln \frac{p(y=i|\mathbf{x})}{p(y=K|\mathbf{x})} = \mathbf{w}_i^{\mathrm{T}}\mathbf{x} + b_i \tag{1.1}$$

所以

$$\frac{p(y=i|\mathbf{x})}{p(y=K|\mathbf{x})} = e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i} \tag{1.2}$$

进一步有

$$\frac{1-p(y=K|\mathbf{x})}{p(y=K|\mathbf{x})} = \sum_{i=1}^{K-1} e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i} \tag{1.3}$$

所以可得

$$p(y=K|\mathbf{x})=\frac{1}{1+\sum_{i=1}^{K-1}e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i}} \tag{1.4}$$

由此可得当$k\neq K$

$$p(y=k|\mathbf{x})=\frac{e^{\mathbf{w}_k^{\mathrm{T}}\mathbf{x}+b_k}}{1+\sum_{i=1}^{K-1}e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i}} \tag{1.5}$$

综上所述，我们有

$$p(y=k|\mathbf{x})=\begin{cases}\frac{e^{\mathbf{w}_k^{\mathrm{T}}\mathbf{x}+b_k}}{1+\sum_{i=1}^{K-1}e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i}} & k\neq K;\\ \frac{1}{1+\sum_{i=1}^{K-1}e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i}} & k=K.\end{cases} \tag{1.6}$$

我们假设总共有m个样例

$$\begin{aligned}&\ell(\mathbf{w},\mathbf{b})\\ =&\sum_{i=1}^{m}\sum_{j=1}^{K}\mathbb{I}(y_i=j)\ln p(y_i=j|\mathbf{x}_i)\\ =&\sum_{i=1}^{m}\sum_{j=1}^{K-1}\mathbb{I}(y_i=j)\ln\frac{e^{\mathbf{w}_j^{\mathrm{T}}\mathbf{x}_i+b_j}}{1+\sum_{t=1}^{K-1}e^{\mathbf{w}_t^{\mathrm{T}}\mathbf{x}_i+b_t}}+\sum_{i=1}^{m}\mathbb{I}(y_i=K)\ln\frac{1}{1+\sum_{t=1}^{K-1}e^{\mathbf{w}_t^{\mathrm{T}}\mathbf{x}_i+b_t}}\\ =&\sum_{i=1}^{m}\sum_{j=1}^{K-1}\mathbb{I}(y_i=j)(\mathbf{w}_j^{\mathrm{T}}\mathbf{x}_i+b_j-\ln(1+\sum_{t=1}^{K-1}e^{\mathbf{w}_t^{\mathrm{T}}\mathbf{x}_i+b_t}))-\sum_{i=1}^{m}\mathbb{I}(y_i=K)\ln(1+\sum_{t=1}^{K-1}e^{\mathbf{w}_t^{\mathrm{T}}\mathbf{x}_i+b_t})\\ =&\sum_{i=1}^{m}\sum_{j=1}^{K-1}\mathbb{I}(y_i=j)(\mathbf{w}_j^{\mathrm{T}}\mathbf{x}+b_j)-\sum_{i=1}^{m}\ln(1+\sum_{t=1}^{K-1}e^{\mathbf{w}_t^{\mathrm{T}}\mathbf{x}_i+b_t})\end{aligned} \tag{1.7}$$

再令$\boldsymbol{\beta}_j=(\mathbf{w}_j;b_j)$以及$\hat{\mathbf{x}}=(\mathbf{x};1)$，可以有

$$\ell(\boldsymbol{\beta})=\sum_{i=1}^{m}\sum_{j=1}^{K-1}\mathbb{I}(y_i=j)(\boldsymbol{\beta}_j^{\mathrm{T}}\hat{\mathbf{x}}_i)-\sum_{i=1}^{m}\ln(1+\sum_{t=1}^{K-1}e^{\boldsymbol{\beta}_t^{\mathrm{T}}\hat{\mathbf{x}}_i}) \tag{1.8}$$

2. 由上一问可得：

$$\begin{aligned}\frac{\partial\ell(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}_j}=&\sum_{i=1}^{m}\mathbb{I}(y_i=j)\hat{\mathbf{x}}_i-\sum_{i=1}^{m}\frac{e^{\boldsymbol{\beta}_j^{\mathrm{T}}\hat{\mathbf{x}}_i}}{1+\sum_{t=1}^{K-1}e^{\boldsymbol{\beta}_t^{\mathrm{T}}\hat{\mathbf{x}}_i}}\hat{\mathbf{x}}_i\\ =&\sum_{i=1}^{m}(\mathbb{I}(y_i=j)-\frac{e^{\boldsymbol{\beta}_j^{\mathrm{T}}\hat{\mathbf{x}}_i}}{1+\sum_{t=1}^{K-1}e^{\boldsymbol{\beta}_t^{\mathrm{T}}\hat{\mathbf{x}}_i}})\hat{\mathbf{x}}_i\\ =&\sum_{i=1}^{m}(\mathbb{I}(y_i=j)-p(y_i=j|\hat{\mathbf{x}}_i))\hat{\mathbf{x}}_i\end{aligned} \tag{1.9}$$

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. 由于正例和负例服从高斯分布，我们用y_1表示正例，用y_2表示负例，所以可知：

$$p(\mathbf{x} = x \mid \mathbf{y} = y_1) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu_1)^{\mathrm{T}}\Sigma^{-1}(x-\mu_1)) \tag{2.1}$$

$$p(\mathbf{x} = x \mid \mathbf{y} = y_2) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu_2)^{\mathrm{T}}\Sigma^{-1}(x-\mu_2)) \tag{2.2}$$

根据样例的分布情况计算$p(\mathbf{y} = y_i)$，可得：

$$p(\mathbf{y} = y_i) = \frac{m_i}{m} = p_i \quad (i = 1, 2) \tag{2.3}$$

根据贝叶斯公式：

$$\begin{aligned} p(\mathbf{y} = y_i \mid \mathbf{x} = x) =& \frac{p(\mathbf{x} = x \mid \mathbf{y} = y_i)p(\mathbf{y} = y_i)}{p(\mathbf{x} = x)} \\ =& \frac{p_i}{p(\mathbf{x} = x)} \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x-\mu_i)^{\mathrm{T}}\Sigma^{-1}(x-\mu_i)) \end{aligned} \tag{2.4}$$

对后验概率取对数，可得（注意到Σ是对称矩阵）

$$\begin{aligned} \ln p(\mathbf{y} = y_i \mid \mathbf{x} = x) =& -\ln p(\mathbf{x} = x) + \ln p_i - \frac{1}{2}(x-\mu_i)^{\mathrm{T}}\Sigma^{-1}(x-\mu_i) \\ =& -\ln p(\mathbf{x} = x) + \ln p_i - \frac{1}{2}(x^{\mathrm{T}}\Sigma^{-1}x + \mu_i^{\mathrm{T}}\Sigma^{-1}\mu_i + 2x^{\mathrm{T}}\Sigma^{-1}\mu_i) \\ =& -\ln p(\mathbf{x} = x) + \ln p_i - \frac{1}{2}x^{\mathrm{T}}\Sigma^{-1}x - \frac{1}{2}\mu_i^{\mathrm{T}}\Sigma^{-1}\mu_i - x^{\mathrm{T}}\Sigma^{-1}\mu_i \end{aligned} \tag{2.5}$$

注意到$\ln p(\mathbf{x} = x)$和$\frac{1}{2}x^{\mathrm{T}}\Sigma^{-1}x$都是和$i$无关的，所以为了方便之后的计算，整理所有和$i$无关的项，用常数$C$代替，可得

$$\ln p(\mathbf{y} = y_i \mid \mathbf{x} = x) = C + \ln p_i - \frac{1}{2}\mu_i^{\mathrm{T}}\Sigma^{-1}\mu_i - x^{\mathrm{T}}\Sigma^{-1}\mu_i \tag{2.6}$$

那么对于给定的x，$\mathbf{y}$取y_1或者y_2的后验概率为

$$\ln p(\mathbf{y} = y_1 \mid \mathbf{x} = x) = C + \ln p_1 - \frac{1}{2}\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1 - x^{\mathrm{T}}\Sigma^{-1}\mu_1 \tag{2.7}$$

$$\ln p(\mathbf{y} = y_2 \mid \mathbf{x} = x) = C + \ln p_2 - \frac{1}{2}\mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2 - x^{\mathrm{T}}\Sigma^{-1}\mu_2 \tag{2.8}$$

现在考虑最优分类平面的定义。如果一个平面将空间分成A，B两个部分，不妨认为落在A部分的x是正例的可能性更大，落在B部分的x是负例的可能性更大。那么当判断x落在A部分时，就

预测它属于正例，反之预测它属于反例，这就满足了最优分类的性质，这样的平面就是最优的分类平面。很显然，当x落在分类平面上时，x属于正例和负例的可能性一样大。根据这一点令$i=1$和$i=2$的后验概率相等来计算最优平面：

$$\begin{aligned} & C+\ln p_1-\frac{1}{2}\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1-x^{\mathrm{T}}\Sigma^{-1}\mu_1=C+\ln p_2-\frac{1}{2}\mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2-x^{\mathrm{T}}\Sigma^{-1}\mu_2 \\ \Leftrightarrow\ & \ln\frac{p_1}{p_2}-\frac{1}{2}(\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1-\mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2)=x^{\mathrm{T}}\Sigma^{-1}(\mu_1-\mu_2) \end{aligned} \tag{2.9}$$

上式是一个超平面方程，由于x的系数为$\Sigma^{-1}(\mu_1-\mu_2)$，所以$\Sigma^{-1}(\mu_1-\mu_2)$是超平面的一个法向量。

现在再考虑LDA。根据LDA的定义，所有样例将要被映射到w上，而$w=S_w^{-1}(\mu_1-\mu_2)$，而$S_w^{-1}=\Sigma_1+\Sigma_2=2\Sigma$。现在只考虑方向，那么$w=\Sigma^{-1}(\mu_1-\mu_2)$就是$LDA$将样本投影到的直线，这也刚好是最优分类平面的法向量。考虑到LDA将所有样本映射到w上，所以需要一个阈值来作为正例和负例的分界。设置阈值为$\ln\frac{p_1}{p_2}-\frac{1}{2}(\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1-\mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2)$，小于阈值预测为正例，反之预测为负例。现在来观察这个分类器是否满足最优分类性质。

由于(2.9)定义了最优平面的方程，所以：

当$\ln\frac{p_1}{p_2}-\frac{1}{2}(\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1-\mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2)>x^{\mathrm{T}}\Sigma^{-1}(\mu_1-\mu_2)$时，样本为正例的可能性大于样本为负例的可能性，所以预测为正例；

当$\ln\frac{p_1}{p_2}-\frac{1}{2}(\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1-\mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2)<x^{\mathrm{T}}\Sigma^{-1}(\mu_1-\mu_2)$时，样本为负例的可能性大于样本为正例的可能性，所以预测为负例。

在LDA分类器中，每一个样本x都被映射为一个标量$x^{\mathrm{T}}\Sigma^{-1}(\mu_1-\mu_2)$，所以：

在$\ln\frac{p_1}{p_2}-\frac{1}{2}(\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1-\mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2)>x^{\mathrm{T}}\Sigma^{-1}(\mu_1-\mu_2)$时预测为正例，根据最优平面的性质，这也是样本为正例可能性大于负例可能性的情况；

在$\ln\frac{p_1}{p_2}-\frac{1}{2}(\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1-\mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2)<x^{\mathrm{T}}\Sigma^{-1}(\mu_1-\mu_2)$时预测为负例，根据最优平面的性质，这也是样本为负例可能性大于正例可能性的情况。

综上所述，LDA方法满足最优分类性质。

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 2，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.

表 2: 算法在测试数据集上泛化性能测试结果，先报告在每个类别上的查全率和查准率，最后报告在整个测试数据集上的准确率。

标记	1	2	3	4	5	准确率
查全率	0.92	0.92	1.00	1.00	0.67	0.91
查准率	0.99	0.80	0.33	0.44	0.26	

实验目的

使用对数几率回归解决多分类问题

实验过程

本次程序的实现主要按照以下几个流程：读入数据，过采样，归一化，进行训练，预测数据。其中关键步骤在于过采样和进行训练。

过采样：

过采样的算法采用了smote算法，也就是针对每一个少数类进行过采样。具体的流程为对于少数类中的每一个样例，寻找他们的k近邻，然后随机挑选一个k近邻，在该样例和近邻之间插值，作为插入的新样本。如此重复，直到少数类数量接近其他类。

训练过程：

因为是多分类问题，所以训练的过程主要采用了One vs Rest的思想，每次把一个类作为正例，其余所有类作为反例，按照二分类问题的方法来训练出一组针对正例类的参数，这样针对每一个类别进行一次训练，共计训练出5个模型。

在训练的过程中，需要首先确定模型的cost function，我所采用的cost function是

$$J = \frac{1}{m}\sum_{i=1}^{m}(-y_i \log(h(\mathbf{x}_i)) + (1-y_i)\log(1-h(\mathbf{x}_i))) \tag{3.1}$$

其中

$$h(\mathbf{x}) = \frac{1}{1+\exp(-\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x})} \tag{3.2}$$

接下来只要对J最小化，求得对应的$\boldsymbol{\theta}$，就完成了对这个类的训练。在此基础上，我为$cost$增加了正则项，来避免过拟合的现象。

具体的优化算法我采用的是梯度下降，大致流程为首先计算出J对于θ的偏导数：

$$\frac{\partial J}{\partial \boldsymbol{\theta}} = \frac{1}{m}\sum_{i=1}^{m}(h(\mathbf{x}_i) - y_i)\mathbf{x}_i \tag{3.3}$$

梯度下降的迭代公式为

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} - \alpha\frac{\partial J}{\partial \boldsymbol{\theta}^{k-1}} \tag{3.4}$$

其中α是下降步长。（由于采用了正则化，所以实际的梯度下降公式多了一项关于正则化的，但是总体思路没有区别）

训练结束得到最终的$\boldsymbol{\theta}$之后，就可以用$\boldsymbol{\theta}$来对测试集做预测了。

预测过程：
考虑到这是一个多分类问题，所以在进行预测的时候实际上需要得到多个预测值然后进行比较。本题是5类，所以根据上面的训练过程会得到5个对应的$\boldsymbol{\theta}$，记为$\mathbf{T} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, ..., \boldsymbol{\theta}^{(5)})$，对于每一个测试样例$(\mathbf{x}; y)$，计算$h_i(\mathbf{x}) = \frac{1}{1+\exp(-\boldsymbol{\theta}^{(i)\mathrm{T}}\mathbf{x})}$，得到$k = arg\max_i h_i(\mathbf{x})$，那么就预测样例$(\mathbf{x}; y)$属于$k$类。

归一化：
考虑到不同特征之间差异较大，有的特征大小可以达到几千，有的不到1，所以进行了归一化的处理，具体操作为针对每一种特征，计算其均值μ和方差σ，然后对每一个特征值t，构造$t' = (t - \mu)/\sigma$，这样不同的特征的范围就大致相同了。

超参数的设置：
超参数主要包括k近邻个数，梯度下降迭代次数，步长等，设置如下表：

表 3: 超参数设置

k(近邻个数)	5
λ(正则化参数)	0.1
α(梯度下降步长)	5
num_iters(梯度下降迭代次数)	1000

实验结果
对于训练集的预测准确度达到了0.94，对于测试集的预测准确度到达了0.91，预测的结果综合来看比较乐观。

实验心得

- 对于少数类别的处理非常重要，提高少数类别的准确率也很不容易。
- 选择不同的优化算法会显著影响结果收敛与否，收敛速度。
- 如何处理不同特征之间范围差异过大的问题也会影响算法的性能。
- matlab程序的编写和c++还是有很大不同，想要灵活熟练运用matlab还需要大量的积累和练习。