机器学习导论
作业二

141110089, 王子豪, wangzihao_fcb@126.com

2018 年 4 月 17 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) I denote* $(\mathbf{x}^{\mathrm{T}}, 1)^{\mathrm{T}}$ *as* $\hat{\mathbf{x}}$ *and* $(\mathbf{w}_i^{\mathrm{T}}, b_i)^{\mathrm{T}}$ *as* θ_i*, thus, we have for each* i

$$\mathbf{w}_i^{\mathrm{T}}\mathbf{x} + b_i = \theta_i^{\mathrm{T}}\hat{\mathbf{x}}$$

.

Consequently

$$\begin{aligned} \ln \frac{p(y=1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \theta_1^{\mathrm{T}}\hat{\mathbf{x}} \\ \ln \frac{p(y=2|\mathbf{x})}{p(y=K|\mathbf{x})} &= \theta_2^{\mathrm{T}}\hat{\mathbf{x}} \\ &\cdots \\ \ln \frac{p(y=K-1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \theta_{K-1}^{\mathrm{T}}\hat{\mathbf{x}} \end{aligned}$$

Therefore, the log-likelihood has the following form :

$$\begin{aligned}
\ell &= log(\prod_{i=1}^{m} p(y_i|\mathbf{x_i}; \mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{K-1}, \mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{K-1})) \\
&= log(\prod_{i=1}^{m} p(y_i|\hat{\mathbf{x}}_\mathbf{i}; \theta_1, \theta_2, \cdots, \theta_{K-1})) \\
&= \sum_{i=1}^{m} log(p(y_i|\hat{\mathbf{x}}_\mathbf{i}; \theta_1, \theta_2, \cdots, \theta_{K-1})) \\
&= \sum_{i=1}^{m} log[\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) \frac{e^{\theta_j^{\mathrm{T}} \hat{\mathbf{x}}}}{\sum_{h=1}^{K-1} e^{\theta_h^{\mathrm{T}} \hat{\mathbf{x}}} + 1} + \mathbb{I}(y_i = K) \frac{1}{\sum_{h=1}^{K-1} e^{\theta_h^{\mathrm{T}} \hat{\mathbf{x}}} + 1}] \\
&= \sum_{i=1}^{m} (log[\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) e^{\theta_j^{\mathrm{T}} \hat{\mathbf{x}}} + \mathbb{I}(y_i = K)] - log[\sum_{h=1}^{K-1} e^{\theta_h^{\mathrm{T}} \hat{\mathbf{x}}} + 1]) \\
&= \sum_{i=1}^{m} [\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) \theta_j^{\mathrm{T}} \hat{\mathbf{x}} + 0] - \sum_{i=1}^{m} log[\sum_{h=1}^{K-1} e^{\theta_h^{\mathrm{T}} \hat{\mathbf{x}}} + 1]
\end{aligned} \tag{1.1}$$

where m is the number of the sample. And note that

$$log(\sum_{j=1}^{K-1} \mathbb{I}(y_i = j) e^{\theta_j^{\mathrm{T}} \hat{\mathbf{x}}} + \mathbb{I}(y_i = K)) = \begin{cases} log(e^{\theta_u^{\mathrm{T}} \hat{\mathbf{x}}}) = \theta_u^{\mathrm{T}} \hat{\mathbf{x}} & if\ j = u \\ log(1) = 0 & if\ j = K \end{cases}$$

(2)

$$\frac{\partial \ell}{\partial \theta_j} = \sum_{i=1}^{m} [\mathbb{I}(y_i = j) \hat{\mathbf{x}} + \frac{e^{\theta_j^{\mathrm{T}} \hat{\mathbf{x}}} \hat{\mathbf{x}}}{\sum_{h=1}^{K-1} e^{\theta_h^{\mathrm{T}} \hat{\mathbf{x}}} + 1}] \tag{1.2}$$

Therefore, $\frac{\partial \ell}{\partial \theta_j}$ is a vector, and consequently, the gradient of the log-likelihood is

$$(\frac{\partial \ell}{\partial \theta_2}, \frac{\partial \ell}{\partial \theta_2}, \cdots, \frac{\partial \ell}{\partial \theta_{K-1}})^{\mathrm{T}}$$

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. *Proof:*

From the analysis about LDA on book, the ideal w has the following form

$$w = S_w^{-1}(\mu_1 - \mu_2)$$

Note that $S_w = \Sigma + \Sigma$,

$$w = (2\Sigma)^{-1}(\mu_1 - \mu 2)$$

Given a x(k-dimensinal), considering the density function of k-dimensional normal distributions $N(\mu_1, \Sigma)$ *and* $N(\mu_2, \Sigma)$, *the probability that x is generated from* $N(\mu_1, \Sigma)$ *is*

$$\frac{1}{(\frac{\pi}{x})^k|\Sigma|}e^{-\frac{1}{2}(x-\mu_1)^{\mathrm{T}}\Sigma^{-1}(x-\mu_1)}$$

and the probability that x is generated from normal distributions $N(\mu_2, \Sigma)$ *and* $N(\mu_2, \Sigma)$, *the probability that x is generated from* $N(\mu_1, \Sigma_1)$ *is*

$$\frac{1}{(\frac{\pi}{x})^k|\Sigma|}e^{-\frac{1}{2}(x-\mu_1)^{\mathrm{T}}\Sigma^{-1}(x-\mu_1)}$$

.

Therefore, the hyperplane corresponding to the "optimal classification" is determined by the following equation

$$-\frac{1}{2}(x-\mu_1)^{\mathrm{T}}\Sigma^{-1}(x-\mu_1) = -\frac{1}{2}(x-\mu_1)^{\mathrm{T}}\Sigma^{-1}(x-\mu_1)$$

$$-\frac{1}{2}x^{\mathrm{T}}\Sigma^{-1}\mu_1 + \frac{1}{2}\mu_1^{\mathrm{T}}\Sigma^{-1}x - \frac{1}{2}\mu_1^{\mathrm{T}}\Sigma^{-1}\mu_1 - \frac{1}{2}x^{\mathrm{T}}\Sigma^{-1}\mu_2 + \frac{1}{2}\mu_2^{\mathrm{T}}\Sigma^{-1}x - \frac{1}{2}\mu_2^{\mathrm{T}}\Sigma^{-1}\mu_2 = 0$$

$$x^{\mathrm{T}}\Sigma^{-1}(\mu_1 - \mu_2) - \frac{1}{2}(\mu_1 + \mu_2)^{\mathrm{T}}\Sigma^{-1}(\mu_1 - \mu_2)$$

According to knowledge from Plane Geometry, the normal vector of the classification hyperplane is parallel to the w in LDA, which means that the classification hyperplane of both methods are parallel to each other. Therefore, these two classification hyperplane are the same. And these two methods are equivalent.

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。

Table 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 4，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.

本次实验报告的结构如下所示。

3.1 实验目的、简介

本次实验是一次机器学习的初学者们初次使用有关算法从数据中获取数据、建立模型的实战。本次实验使用网络上的UCI数据集处理多分类问题，目的是通过Logistic Regression的方法，学习数据集中数据5个不同类别与其10个不同的特征（特征为连续实数）之间的联系，并且构建对新的样本点具有分类功能的分类器，实现预测。

本次实验的编程语言是Python3.

3.2 数据预处理

3.2.1 数据的导入

本次实验的原始数据集的格式类型txt文本文档，利用Python3中内置的函数readline，我自定义函数txt2df：该函数需要数据变量为txt格式的数据特征集以及数据标签集，返回一个将特征集与标签集对应在一起的Python DataFrame。

经过检查发现，本实验所用数据有4935条数据作为训练集，539条数据作为测试集，每条训练集与测试集数据含有不算常数项在内的10个特征，训练集与测试集都含有五种不同种类标记的数据。

每条数据均完整，不需要额外进行填补缺失值的工作。

3.2.2 数据的标准化

我们知道，在线性回归中不同特征取值的数量级对于确定模型参数有着重要的影响，取值数量级较大的特征天然地对于模型结果有着更大地影响。通过观察，我发现特征3与特征9的数量级远远大于其他，因此如果直接用原始数据进行Logistics Regression,得到的模型受到特征3，9的影响巨大。为了避免这一点，我自定义的数据标准化函数normalize来对数据进行标准化。

我采用的标准化方法即正太分布标准化法(超参数h_1:normalized = True，False 是否标准化数据)，对于每个训练集上的特征F，我将每个样本点的该特征减去这一特征的均值，再除以这个特征的标准差，得到标准化后后的特征。

$$\hat{F}_i = \frac{F_i - \mu_F}{\sigma_F}$$

同时为了保证训练集、测试集的对应关系，我将这个特征的均值与标准差记录下来，在预测时以同样的方式作用于新数据上。

3.3 构建多分类算法

我们知道Logistic Regression是一种经典的二分类算法，但是二分类算法稍加改进，便可用于多分类问题中。书中提出了三种改进方法：“一对其余”、“一对一”、“多对多”。这里，我才用“一对其余”的方法利用Logistic Regression实现多分类，其余的改进方式可在以后尝试。

由于本实验要解决一个五分类问题，因此在应用“一对其余”策略改进Logistics Regression时，要分别拟合出五个分类器：C_1, C_2, C_3, C_4, C_5,其中C_i是将有第i类标记标记作为正例，将没有第i类标记作为反例，训练得到的分类器。

3.3.1 单个分类器的构建，以C_1为例

由上述约定，分类器C_1是将有标记1数据作为正例，没有标记1样本作为反例训练得到的分类器。在训练该分类器时，应给数据附加新的标记，如将带有原始标记1的数据标记为1，将未带有原始标记1的数据标记为0.

设训练集样本数为m，我们做如下约定：训练集特征矩阵记为$X_{m\times 11}$(11为样本的特征数，包含常数项1),训练集标记矩阵记为$Y_{m\times 1}$,待优化权重记为$w_{11\times 1}$, Sigmoid函数记为$\sigma(x)$. 我们优化权重的目标心里（即损失函数）为

$$min||\sigma(Xw)-Y||$$

,优化算法为梯度下降法（超参数h_2,learning rate; 超参数h_3 number of iterations）

梯度下降法分析：

$$\frac{\partial||\sigma(Xw)-Y||}{\partial w}=X^{\mathrm{T}}(|\sigma(Xw)-Y|)$$

$$\Rightarrow w=w-h_2*\frac{\partial||\sigma(Xw)-Y||}{\partial w}=w-h_2*X^{\mathrm{T}}(|\sigma(Xw)-Y|)$$

基于此数学推导，我自定义了函数fit，其对于任意初始化参数，按照上式中的梯度下降方式，依次迭代优化权重，并且画出训练集与测试集上的$||\sigma(Xw)-Y||$的下降趋势与h_3的关系图。最后返回最终的权重weight，以及在训练集与测试机上损失函数的随着梯度下降的变化。

3.3.2 解决类别不平衡问题，以SMOTE算法为例

在训练上述单个分类器时，我发现每个分类器的训练集的正反例样本个数常常差距巨大，例如以C_1,该分类器的训练样本中4431个为正例，却只有504个反例，一个将全部样本预测为正例的分类器可以在训练样本上达到0.85的正确率，为了避免这种情况，我们通常可以使用“过采样”、“欠采样”、“阈值移动”等方式修正。我将使用“过采样”方式中的经典算法SMOTE，对于每个子分类器优化。

SMOTE算法对于少数类的每个样本x_i，取出x_i的K临近少数类样本，再从K临近少数类样本中随机选出一个样本，与x_i随机差值得到新的样本（超参数h_4 K）,循环进行该步骤直到少数类样本数与多数类样本数大体一致。

基于此，我自定了smote函数，该函数输入两类样本，并返回两个dataframe，保持多数类不变并且对少数类进行差值扩充。在设计smote函数时，通过遍历10以内的K发现其数值对结果并无巨大影响，因此我先验地将K设置为3。

3.3.3 多个子分类器的打分

设五个分类器的权重分别为w_1,w_2,w_3,w_4,w_5,对于测试集新样本$x^{'}$,决策函数如下

$$D(x)=(i \quad s.t. \quad \sigma(\mathbf{w}_i^T x) \quad max)$$

3.4 超参数分析结果展示

由上述分析，我一共假设了四个超参数，但由遍历结果可知，超参数h_4对于结果影响不大。

3.4.1 是否标准化

以$learningrate=0.001$, $n_ieration=200$ 为例

可以看出对这对参数而言，做标准化后，分类器准确率以及子分类器查全率、查重率全面提高，因此做标准化对于优化分类非常重要。其他参数组合情况类似。

Table 2: 做标准化

标记	1	2	3	4	5	准确率
查全率	0.94	0.92	0.67	0.75	0.50	0.92
查准率	0.98	0.74	0.22	0.38	0.33	

Table 3: 不做标准化

标记	1	2	3	4	5	准确率
查全率	0.90	0.92	1	0	0.25	0.88
查准率	0.98	0.47	0.5	0.0	0.2	

3.4.2 学习率与迭代次数

$learningrate = 0.001$, $n_ieration = 200$

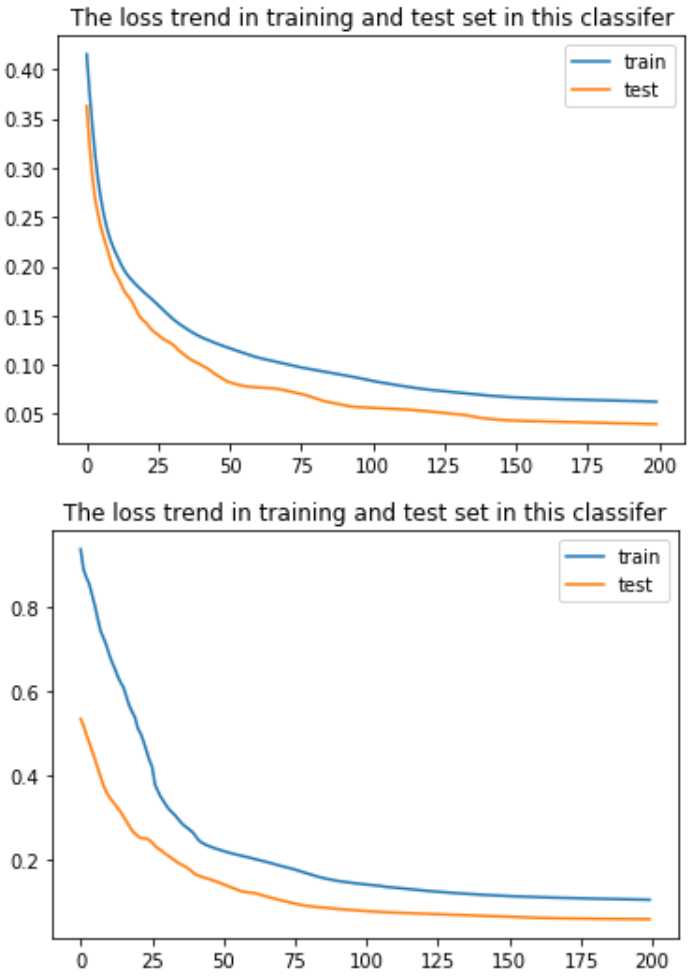

Figure 1: the cost picture

$learningrate = 0.01$, $n_ieration = 200$

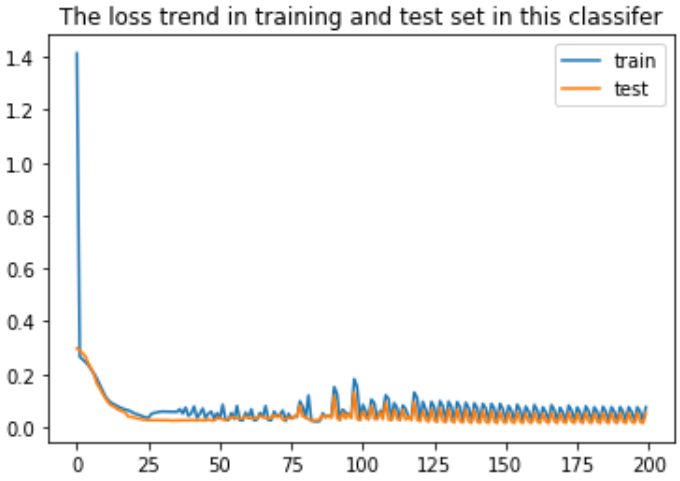

Figure 2: the cost picture

$learningrate = 0.0001$, $n_ieration = 600$

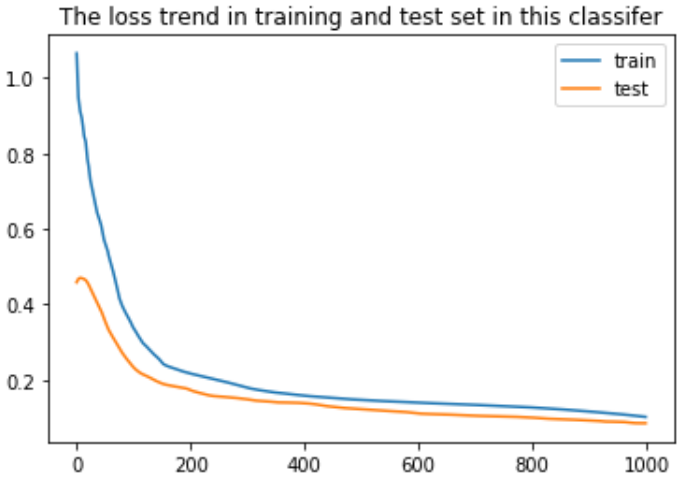

Figure 3: the cost picture

学习率与迭代次数是相关的，我先通过遍历确定合适的学习率，再根据此选择合适的迭代次数。

我对学习率按照[0.1,0.03,0.01,0.003,0.001,0.0001]进行遍历，发现当学习率$r \geq 0.01$，子分类器在十几次次迭代后便快速收敛，有时甚至出现先收敛后波动的情况，当学习率$r \leq 0.001$，子分类器需要长达800次迭代以上才会收敛；综上，学习率数量级应在$\frac{1}{1000}$左右。固定学习率为0.001，对于每个子分类器而言，当迭代次数达到100 左右时，出现明显的收敛趋势，且在迭代次数达到200时基本收敛，因此我将最大的迭代次数设置为200，避免不必要的计算开销或者过拟合。最终结果如下表所示。

Table 4: 结果

标记	1	2	3	4	5	准确率
查全率	0.94	0.92	0.67	0.75	0.50	0.92
查准率	0.98	0.74	0.22	0.38	0.33	