机器学习导论
作业二

2018 年 6 月 30 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，

$$\begin{aligned}
\ln \frac{p(y=1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_1^{\mathrm{T}}\mathbf{x} + b_1 \\
\ln \frac{p(y=2|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_2^{\mathrm{T}}\mathbf{x} + b_2 \\
&\cdots \\
\ln \frac{p(y=K-1|\mathbf{x})}{p(y=K|\mathbf{x})} &= \mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x} + b_{K-1}
\end{aligned} \tag{1.1}$$

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) 样本属于各类的概率之和为1:

$$\sum_{c=1}^{c=K} p(y=c|\mathbf{x}) = 1 \tag{1.2}$$

与式(1.1)联立得：

$$\begin{aligned} p(y=1|\mathbf{x}) &= \frac{e^{\mathbf{w}_1^{\mathrm{T}}\mathbf{x}+b_1}}{1+\sum_{c=1}^{K-1} e^{\mathbf{w}_c^{\mathrm{T}}\mathbf{x}+b_c}} \\ p(y=2|\mathbf{x}) &= \frac{e^{\mathbf{w}_2^{\mathrm{T}}\mathbf{x}+b_2}}{1+\sum_{c=1}^{K-1} e^{\mathbf{w}_c^{\mathrm{T}}\mathbf{x}+b_c}} \\ &\cdots \\ p(y=K-1|\mathbf{x}) &= \frac{e^{\mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x}+b_{K-1}}}{1+\sum_{c=1}^{K-1} e^{\mathbf{w}_c^{\mathrm{T}}\mathbf{x}+b_c}} \\ p(y=K|\mathbf{x}) &= \frac{1}{1+\sum_{c=1}^{K-1} e^{\mathbf{w}_c^{\mathrm{T}}\mathbf{x}+b_c}} \end{aligned} \tag{1.3}$$

记$\beta_j = (\mathbf{w}_j; b)$, $\theta = (\beta_1, \beta_2, \ldots, \beta_{K-1})$, “对数似然”*(log-likelihood)*函数可表示为：

$$l(\theta) = \sum_{i=1}^{m} \ln p(y_i|\mathbf{x}_i; \theta) \tag{1.4}$$

记$y_{ic} = \mathbb{I}(y_i = c)$,式(1.4)中的似然项可以重写为：

$$p(y_i|\mathbf{x}_i; \theta) = \prod_{c=1}^{K} [p(y_i = c|\mathbf{x}_i; \theta)]^{y_{ic}} \tag{1.5}$$

把式(1.5)代入式(1.4)并根据式(1.3)得到：

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{m}\sum_{c=1}^{K} y_{ic} \ln p(c|\mathbf{x}_i; \theta) \\ &= \sum_{i=1}^{m}[\sum_{c=1}^{K-1}(y_{ic}\beta_c^T\hat{x}_i - \ln(1+\sum_{j=1}^{K-1} e^{\beta_j^T\hat{x}_i})) - y_{iK}\ln(1+\sum_{j=1}^{K-1} e^{\beta_j^T\hat{x}_i})] \\ &= \sum_{i=1}^{m}[(\sum_{c=1}^{K-1} y_{ic}\beta_c^T\hat{x}_i) - \ln(1+\sum_{j=1}^{K-1} e^{\beta_j^T\hat{x}_i})] \end{aligned} \tag{1.6}$$

其中，$\hat{x}_i = (\mathbf{x}_i; 1)$.

(2) 式(1.6)求导：

$$\frac{\partial l(\theta)}{\partial \beta_c} = \sum_{i=1}^{m} \hat{x}_i(y_{ic} - p(c|\hat{x}_i; \theta)) \tag{1.7}$$

for $c = 1, 2, \ldots, K-1$.

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”：若分类器在得到测试样例在不同类别的分类概率后，取概率最大的类别作为最终预测的类别输出，则满足“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)方法可以达到“最优分类”。（提示：找到定义的最优分类的分类平面。）

Solution. 此处用于写解答*(*中英文均可*)* 高斯分布的概率密度函数为：

$$p(\mathbf{x}) = (2\pi)^{-d/2}|\Sigma|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right) \tag{2.1}$$

因此当$\Sigma_1 = \Sigma_2 = \Sigma$时，任意样本属于正例或者负例的概率只与$\mu_1, \mu_2$有关：

$$\begin{aligned} p(y=1|\mathbf{x}) &\propto \exp\left((\mathbf{x}-\boldsymbol{\mu_1})^T(\mathbf{x}-\boldsymbol{\mu_1})\right) \\ p(y=2|\mathbf{x}) &\propto \exp\left((\mathbf{x}-\boldsymbol{\mu_2})^T(\mathbf{x}-\boldsymbol{\mu_2})\right) \end{aligned} \tag{2.2}$$

样本到哪一类中心的距离越小，则采样于该类的概率越大。又因为*LDA*模型中样本的预测类别为距离样本投影点最近的投影中心对应的类别：

$$\begin{aligned} \hat{p}(y=1|\mathbf{x}) &\propto \left((\mathbf{w}^T\mathbf{x}-\mathbf{w}^T\boldsymbol{\mu_1})^T(\mathbf{w}^T\mathbf{x}-\mathbf{w}^T\boldsymbol{\mu_1})\right) \propto \left((\mathbf{x}-\boldsymbol{\mu_1})^T(\mathbf{x}-\boldsymbol{\mu_1})\right) \\ \hat{p}(y=2|\mathbf{x}) &\propto \left((\mathbf{w}^T\mathbf{x}-\mathbf{w}^T\boldsymbol{\mu_2})^T(\mathbf{w}^T\mathbf{x}-\mathbf{w}^T\boldsymbol{\mu_2})\right) \propto \left((\mathbf{x}-\boldsymbol{\mu_2})^T(\mathbf{x}-\boldsymbol{\mu_2})\right) \end{aligned} \tag{2.3}$$

即样本到哪一类中心的距离越小，则预测为该类的概率越大。所以线性判别分析 *(LDA)*方法可以达到“最优分类”。

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。

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

表 1: Page Blocks数据集中每个类别的样本数量。

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/`上能够方便的制作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。

4 实验报告

1. 算法要点

- 模型 Logistic Regression Multi-classification
- 优化方法 Mini-batch Gradient Descendant
- 多分类策略 OvR

2. 过采样方法

此处借鉴somote方法但是和smote方法不同：先计算得出小类的样本中心点x_{center},然后随机选取$K(K$ 为超参数)个小类样本计算其均值x_{kmean},选择一组0到1之间的随机数ζ,利用如下公式的到新的采样：

$$x_{sample}[i] = \zeta[i]x_{center}[i] + (1-\zeta)[i]x_{kmean}[i] \tag{4.1}$$

其中索引i代表对应项的第i个维度。这样做主要是希望能够使新的采样不太偏离中心同时能利用任意K个样本而不要求这K个样本保持邻近的关系。

3. 附加题设置

考虑有一个明显的大类(majority class),因此利用教材的“阈值移动”(threshold-moving)方法对第一类的输出概率进行折减，折减系数为F(超参数)，然后去最大概率的类别。

4. 超参设置

raw 代表未采用任何方法处理样本不平衡问题。

	epoch	batch size	learning rate	K	F
raw	1000	32	0.02	/	/
oversample	1000	32	0.02	5	/
move-threshold	1000	32	0.02	/	0.4

表 2: hyper-parameter

5. 实验结果

标记	1	2	3	4	5	准确率
查全率	0.990	0.784	0.667	1.0	0.25	0.957
查准率	0.964	0.906	1.0	1.0	0.6	

表 3: raw

标记	1	2	3	4	5	准确率
查全率	0.985	0.892	0.667	1.0	0.417	0.965
查准率	0.975	0.917	1.0	0.8	0.625	

表 4: oversample

标记	1	2	3	4	5	准确率
查全率	0.983	0.919	0.667	1.0	0.333	0.963
查准率	0.975	0.871	1.0	1.0	0.571	

表 5: move-threshold

对实验结果进行分析：

- 总体准确率与小类准确率上，过采样方法最好。
- 对第一类的概率进行折减提高了大类的准确率，降低了其召回率，因为大类的测试样本较多，所以总体上也提高了准确率。
- 从中间结果看，预测的误差主要来源于把第2和5类预测为第1类，因此如果对于多分类能同时对各个类别进行阈值移动也许能够提高模型性能，但是这样做超参数太多。

6. 其他

- 虽然训练过程有随机数，但是算法很稳定，每次运行结果基本一致。
- 训练前对数据进行了归一化，一方面避免python3溢出，另一方面提高了训练的效率和质量。
- 第一、三组实验可以五个分类器同时训练，但是为了保持三个个实验统一接口并且此次实验训练时间较短，所以利用for循环进行了五个分类器的训练。