机器学习导论
作业二

151250189, 翟道京, zhaidj@smail.nju.edu.cn

2018 年 5 月 4 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) 考虑该多分类问题满足提示 1 所示 $K-1$ 个对数几率，则

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

根据

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

求和得到

$$p(y=K|\mathbf{x}) = \frac{1}{1+e^{\mathbf{w}_1^{\mathrm{T}}\mathbf{x}+b_1}+e^{\mathbf{w}_2^{\mathrm{T}}\mathbf{x}+b_1}+\cdots+e^{\mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x}+b_{K-1}}}$$

因此有

$$p(y=i|\mathbf{x})) = \begin{cases} \frac{e^{\mathbf{w}_i^{\mathrm{T}}\mathbf{x}+b_i}}{1+e^{\mathbf{w}_1^{\mathrm{T}}\mathbf{x}+b_1}+e^{\mathbf{w}_2^{\mathrm{T}}\mathbf{x}+b_1}+\cdots+e^{\mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x}+b_{K-1}}} & i=1,\ 2,\ 3,\ldots K-1. \\ \frac{1}{1+e^{\mathbf{w}_1^{\mathrm{T}}\mathbf{x}+b_1}+e^{\mathbf{w}_2^{\mathrm{T}}\mathbf{x}+b_1}+\cdots+e^{\mathbf{w}_{K-1}^{\mathrm{T}}\mathbf{x}+b_{K-1}}} & i=K \end{cases}$$

对于多分类问题，为简单起见，这里引入一些记号。令 $\theta_i = (\mathbf{w}_i; b), \theta = (\theta_1^T; \theta_2^T; \ldots; \theta_{K-1}^T; 1), \hat{\mathbf{x}} = (\mathbf{x}; 1)$，此时 $\mathbf{w}_i^T\mathbf{x}+b_i = \theta_i^T\hat{\mathbf{x}}$；对给定数据集 $\{(\mathbf{x}_i; y_i)_{i=1}^m\}$，估计函数 *(hypothesis function)* $h_\theta(\mathbf{x})$ 形式如下所示

$$h_\theta(\mathbf{x}_i) = \begin{bmatrix} p(y_i=1|\mathbf{x}_i;\theta) \\ p(y_i=2|\mathbf{x}_i;\theta) \\ \cdots \\ p(y_i=K-1|\mathbf{x}_i;\theta) \\ p(y_i=K|\mathbf{x}_i;\theta) \end{bmatrix} = \frac{1}{1+\sum_{j=1}^{K-1} e^{\theta_j^T\mathbf{x}_i}} \begin{bmatrix} e^{\theta_1^T\hat{\mathbf{x}}_i} \\ e^{\theta_2^T\hat{\mathbf{x}}_i} \\ \cdots \\ e^{\theta_{K-1}^T\hat{\mathbf{x}}_i} \\ 1 \end{bmatrix}$$

因此该对率回归模型的“对数似然”*(log-likelihood)* 为

$$\ell(\theta) = \sum_{i=1}^{m}\left(\sum_{j=1}^{K-1}\mathbb{I}(y=j)\ln\frac{e^{\theta_j^T\hat{\mathbf{x}}_i}}{1+\sum_{j=1}^{K-1}e^{\theta_j^T\hat{\mathbf{x}}_i}} + \mathbb{I}(y=K)\ln\frac{1}{1+\sum_{j=1}^{K-1}e^{\theta_j^T\hat{\mathbf{x}}_i}}\right)$$

为简便起见，记 $\theta_K^T = 0$，即我们人为添加一个自变量 θ_K^T，此时自然有 $e^{\theta_K^T\hat{\mathbf{x}}} = 1$ 成立。此时对数似然简化为

$$\ell(\theta) = \sum_{i=1}^{m}\sum_{j=1}^{K}\mathbb{I}(y=j)\ln\frac{e^{\theta_j^T\mathbf{x}_i}}{\sum_{j=1}^{K}e^{\theta_j^T\mathbf{x}_i}}$$

(2) 下面计算该“对数似然”*(log-likelihood)* $\ell(\theta)$ 的梯度，$\nabla\ell(\theta)$ 是一个向量，其中 $j \neq K$ 时，第 j 个分量 $\nabla_{\theta_j}\ell(\theta)$ 为

$$\begin{aligned} \nabla_{\theta_j}\ell(\theta) &= \sum_{i=1}^{m}\mathbb{I}(y=j)\nabla_{\theta_j}\left(\ln\frac{e^{\theta_j^T\mathbf{x}_i}}{\sum_{j=1}^{K}e^{\theta_j^T\mathbf{x}_i}}\right) \\ &= \sum_{i=1}^{m}\mathbf{x}_i\mathbb{I}(y=j)\left(1-\frac{e^{\theta_j^T\mathbf{x}_i}}{\sum_{j=1}^{K}e^{\theta_j^T\mathbf{x}_i}}\right) \\ &= \sum_{i=1}^{m}\mathbf{x}_i\left(\mathbb{I}(y=j)-p(y=j|\mathbf{x}_i)\right) \end{aligned}$$

即综上所述

$$\nabla_{\theta_j}\ell(\theta) = \begin{cases} \sum_{i=1}^{m}\mathbf{x}_i\left(\mathbb{I}(y=j)-p(y=j|\mathbf{x}_i)\right),\ as\ j\neq K, \\ 0,\ as\ j=K. \end{cases}$$

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 μ_1, Σ_1 及 μ_2, Σ_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。）

证明. 下面我们从“最优分类”性质靠近 LDA 方法分析。[1]
对于正例和负例服从多变量高斯分布 $\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)$，密度函数满足

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

题目中所描述分类器为“贝叶斯最优分类器”[2]，即对于每个样本，选择能使后验概率 $P(c|\mathbf{x})$ 最大的类别标示。

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

基于贝叶斯定理，$P(c|\mathbf{x})$ 可写为

$$P(c|\mathbf{x}) = \frac{P(c)P(\mathbf{x}|c)}{P(\mathbf{x})}$$

其中 $P(c)$ 为先验概率；$P(\mathbf{x}|c)$ 是样本 $\mathbf{x}$ 相对于类标示记 c 的类条件概率，$P(\mathbf{x})$ 是用于归一化的“证据”因子，对于给定的样本 $\mathbf{x}$，证据因子 $P(\mathbf{x})$ 与类标记无关，因此我们的任务为训练数据 D 来估计先验 $P(c)$ 和似然 $P(\mathbf{x}|c)$，分类器转化为

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

下面我们为了简化问题，分别用 $p(\mathbf{x}|c)$ 代替 $P(\mathbf{x}|c)$ 参与估计，并取对数进行计算

$$\begin{aligned}
h^*(x) &= \arg\max_{c\in\mathcal{Y}} P(c|\mathbf{x}) \\
&= \arg\max_{c\in\mathcal{Y}} p(\mathbf{x}|c)P(c) \\
&= \arg\max_{c\in\mathcal{Y}} \ln\left(p(\mathbf{x}|c)P(c)\right) \\
&= \arg\max_{c\in\mathcal{Y}} \left[-\ln\sqrt{(2\pi)^d det(\Sigma)} - \frac{1}{2}(\mathbf{x}-\mu_c)^T\Sigma^{-1}(\mathbf{x}-\mu_c) + \ln P(c)\right] \\
&= \arg\max_{c\in\mathcal{Y}} \left[-\frac{1}{2}(\mathbf{x}-\mu_c)^T\Sigma^{-1}(\mathbf{x}-\mu_c) + \ln P(c)\right]
\end{aligned}$$

展开上式，考虑

$$-\frac{1}{2}(\mathbf{x}-\mu_c)^T\Sigma^{-1}(\mathbf{x}-\mu_c) = \mathbf{x}^T\Sigma^{-1}\mu_c - \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c - \frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}$$

[1] 部分参考STATS 202: Data mining and analysis. Stanford
[2] 周志华, 机器学习. 第七章: 贝叶斯分类器, P147-P149

得到分类器

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

定义判别函数为

$$\delta_c = \mathbf{x}^T\Sigma^{-1}\mu_c - \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c + \ln P(c)$$

分类器即为

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

对于题目中所示二元高斯分布，两个类别决策边界为

$$\{x : \delta_1(\mathbf{x}) = \delta_2(\mathbf{x})\}$$

即满足

$$\ln\frac{P(c=1)}{P(c=2)} - \frac{1}{2}(\mu_1+\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) + \mathbf{x}^T\Sigma^{-1}(\mu_1-\mu_2)$$

此函数对应投影方向 ω^* 为

$$\omega^* = \Sigma^{-1}(\mu_1-\mu_2).$$

根据课本 (3.4) 节介绍，线性判别方法 (LDA) 获得的最有投影方向为

$$\omega = S_\omega^{-1}(\mu_1-\mu_2).$$

其中

$$S_\omega^{-1} = \Sigma_1 + \Sigma_2 = 2\Sigma$$

显然我们可以发现 ω^* 与 ω 方向相同，实际上认为 $\omega^* = \omega$。因而在题目研究条件下，线性判别分析 (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 **??**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. 我使用 MATLAB 完成本次编程作业，实现了题目 (2) 中所要求的实验功能。为了方便调用，我把所有的函数都放在了 main 文件中，但是由于我使用的 MAC OS，读取文件似乎和 WIndows 有一些区别，所以还需要助教学长学姐手动将数据集加入路径:）
我的程序目录如下所示

1. LR_main.m：主程序。
2. MCLR：函数，Multi-class logit regression with Newton Method；
3. predictOneVsAll：函数，预测测试集结果。
4. Sigmoid: sigmoid 函数，自变量 $\mathbf{x}$ 为矩阵；
5. sigmoid: sigmoid 函数，自变量 x 为实数。
6. Grad: 函数，求梯度。
7. SMOTE：函数，当正例数目小于 5% 时进行增益，使正例数目多于 10%。
8. accuracy：函数，输出结果查重率与查准率。

程序实现步骤如下：

Step 0 程序初始化。

Step 1 读入数据。分别记为为"test_feature","test_label","train_feature","train_label".

Step 2 OVR 学习。我们共需要训练 5 个分类器，每个分类器参与训练时，将该类样例作为正例，其他类样例作为反例进行训练。考虑到一些类的样本数目较少，我们在处理该类处理器时选择了 oversampling 方式，编写了 SMOTE 函数。每次分类时若正例数目 <5%，我们通过 SMOTE 增益，将其数目增益到 10% 之上。后面附录中我给出了 SMOTE 的算法。我们选择牛顿法法求解 cost function 的最优解，我们最大迭代次数选择为 max_iter = 5000 ，精确要求选择为 $err = 1e-10$。Step2 结束后，我们获得了 N 个分类器，分类器信息存储在矩阵 *all_beta* 中。

Step 3 使用"predictOneVsAll.m"函数预测测试集结果，通过"accuracy.m"函数输出每一个标记的查全率与查准率。同时输出准确率。

下表格是我在"过采样"前后的实验结果，其中"过采样"后，选取某次结果进行展示。

表 2: 未经"过采样"(oversampling) 时的测试结果

标记	1	2	3	4	5	准确率
查全率	0.967	0.886	NAN	1.000	0.500	0.957
查准率	0.989	0.838	0.00	1.000	0.250	

表 3: 经“过采样”(oversampling) 后的测试结果

标记	1	2	3	4	5	准确率
查全率	0.984	0.939	0.00	0.016	0.375	0.860
查准率	0.888	0.838	0.00	0.25	0.250	

下面大致介绍了我的程序中 SMOTE 算法的思想。

Result: Write here the result
计算当前正例所占参加训练样本比例 rate;
if *rate* $>$ *5%* **then**
    不增益正例数目;
    退出程序;
else
    while *rate* $<$ *10%* **do**
        对每一个正例 i，挑选距离该正例欧式距离最近的 5 组;
        在 5 组邻居内随机挑选一组正例 j，在 i 与 j 连线上随机插入一点，从而每次循环过后正例数翻倍;
        更新此时的正例数与正例占比 rate;
    end
end

Algorithm 1: SMOTE 算法

根据表 2 与表 3 的对比，我的 oversampling 效果不是很好... 在这里特别感谢浦善文同学与我关于 SMOTE 算法的讨论，我们都选用了欧式距离最近的五个邻居进行插值，他的插值方式参数更多，更加精密，我的插值方式注重简洁性。