机器学习导论
作业二

141110091, 吴璐欢, lhwunju@outlook.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) According to the hint 1, let* $\hat{\mathbf{x}} = (\mathbf{x}, 1), \boldsymbol{\beta}_k = (\mathbf{w}_k; \mathbf{b}_k), k = 1, \cdots, K-1$*, so we have the assumption for multi-class logistic regression problem:*

$$\ln \frac{p(y=k|\hat{\mathbf{x}})}{p(y=K|\hat{\mathbf{x}})} = \boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}, \quad \textit{for } k = 1, \cdots, K-1$$

$$\Rightarrow p(y=k|\hat{\mathbf{x}}) = e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}} p(y=K|\hat{\mathbf{x}}), \quad \textit{for } k = 1, \cdots, K-1 \tag{1.1}$$

Since probabilities of all classes sum up to 1, i.e.

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

Substituting (1.1) into (1.2), we have

$$p(y=k|\hat{\mathbf{x}}) = \begin{cases} \frac{e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}}}{1+\sum_{k=1}^{K-1} e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}}} & k=1,\cdots,K-1; \\ \frac{1}{1+\sum_{k=1}^{K-1} e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}}} & k=K. \end{cases} \tag{1.3}$$

The log-likelihood is:

$$\ell(\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_{K-1}) = \sum_{i=1}^{m} \ln p(y_i|\hat{\mathbf{x}}_i,\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_{K-1}) \tag{1.4}$$

Using hint 2 and (1.3), we could rewrite the log-likelihood for the i-th example as:

$$\begin{aligned} \ln p(y_i|\hat{\mathbf{x}}_i,\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_{K-1}) &= \ln[(\sum_{k=1}^{K-1}\mathbb{I}(y_i=k)\frac{e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i}}{1+\sum_{k=1}^{K-1}e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i}}) + \mathbb{I}(y_i=K)\frac{1}{1+\sum_{k=1}^{K-1}e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i}}] \\ &= \ln[(\sum_{k=1}^{K-1}\mathbb{I}(y_i=k)e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i}) + \mathbb{I}(y_i=K)] - \ln(1+\sum_{k=1}^{K-1}e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i}) \\ &= (\sum_{k=1}^{K-1}\mathbb{I}(y_i=k)\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i) + \mathbb{I}(y_i=K)(1-\mathbb{I}(y_i=K)) - \ln(1+\sum_{k=1}^{K-1}e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i}) \end{aligned} \tag{1.5}$$

Therefore, the log-likelihood (1.4) could be rewritten as:

$$\ell(\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_{K-1}) = \sum_{i=1}^{m}[(\sum_{k=1}^{K-1}\mathbb{I}(y_i=k)\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i) + \mathbb{I}(y_i=K)(1-\mathbb{I}(y_i=K)) - \ln(1+\sum_{k=1}^{K-1}e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i})] \tag{1.6}$$

(2) For $k=1,\cdots,K-1$,

$$\begin{aligned} \frac{\partial\ell(\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_{K-1})}{\partial\boldsymbol{\beta}_k} &= \sum_{i=1}^{m}\hat{\mathbf{x}}_i\mathbb{I}(y_i=k) - \frac{\hat{\mathbf{x}}_i e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i}}{1+\sum_{k=1}^{K-1}e^{\boldsymbol{\beta}_k^{\mathrm{T}}\hat{\mathbf{x}}_i}} \\ &= \sum_{i=1}^{m}\hat{\mathbf{x}}_i[(\mathbb{I}(y_i=k) - p(y_i=k|\hat{\mathbf{x}}_i;\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_{K-1})] \end{aligned} \tag{1.7}$$

The gradient is $(\frac{\partial\ell(\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_{K-1})}{\partial\boldsymbol{\beta}_1},\cdots,\frac{\partial\ell(\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_{K-1})}{\partial\boldsymbol{\beta}_{K-1}})^{\mathrm{T}}$.

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 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. *Let* $\mathbf{X}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathbf{X}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$*, then the respective probabilities distribution functions are:*

$$f_i(x) = \frac{1}{\sqrt{(2\pi)^k |\boldsymbol{\Sigma}_i|}} \exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_{\boldsymbol{i}}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)) \qquad i = 1, 2,$$

where k *is the dimension of the feature space.*

According to Page 62 in the textbook , in LDA, the projection line $\mathbf{w}$ *is given by:*

$$\mathbf{w} = \mathbf{S}_{\mathbf{w}}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2), \tag{2.1}$$

where $\mathbf{S}_{\mathbf{w}} = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2$.

The optimal classification hyper-plane is given by:

$$f_1(\mathbf{x}) = f_2(\mathbf{x}). \tag{2.2}$$

When $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$*, (2.1) is equivalent to:*

$$\mathbf{w} = \frac{1}{2}\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2), \tag{2.3}$$

and (2.2) is equivalent to:

$$(\mathbf{x} - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_1) = (\mathbf{x} - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_2)$$

$$\Rightarrow \qquad 2\mathbf{x}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_2 = 0 \tag{2.4}$$

From equations (2.3) and (2.4) , we could see that the projection line in LDA is normal to the optimal classification hyper-plane. Therefore, LDA achieves "optimal classification" in the case when $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$.

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。

实验报告.

1. General Procedures

   (1) Read data: $\mathbf{x}, \mathbf{y}$. Generally, notate the feature matrix by $\mathbf{x}$ and the label vector by $\mathbf{y}$.

   (2) Normalize feature matrix in both training set and test set:
   $\mathbf{x} = \frac{\mathbf{x}-\mu}{\sigma}$, where μ is the mean of $\mathbf{x}$, and σ the standard deviation of $\mathbf{x}$.

   (3) Input data into **One-vs-Rest** algorithm
   Suppose $K =$ the number of labels available.

   Step 1. For each possible value of label, i.e. for $k = 1 : K$

   (i) Re-label the training set according to $y = k$ or not. that is:
   if $y = k$, relabel y as 1; otherwise relabel y as 0.

   (ii) Feed the relabeled training-set into **Binary-Logistic-Regression** which outputs a model (the weight matrix in the formula of logistic regression).

   (iii) Make predictions on test set:
   Use the trained model to obtain the predicted probabilities that each sample belongs to class k, and store them
   Obtain predicted labels: predict 1 (belongs to class k) if probability ≥ 0.5, otherwise 0 (not belongs to class k).
   Use the predicted labels and probabilities to calculate the values of Recall and Precision, and 2-norm Error.

   Step 2. Predict classes on test set:
   For each sample in the test set, let it's predicted class be the one with maximum predicted probability that it belongs to class k.

   Step 3. Calculate the value of **Accuracy** based on the predicted classes and the ground truth.

2. Description of Algorithms

   (1) **One-vs-Rest** is introduced in the General Procedures above.

   (2) **Binary-Logistic-Regression**:
   Input: feature matrix $\mathbf{x}$, label vector $\mathbf{y}$ (0-or-1 value) of training set, learning-rate and maximum-iterations
   Output: the weight matrix for logistic regression formula

   Step 1. $\mathbf{x}, \mathbf{y} =$ **SMOTE(x, y, k)**: Use **k**-nearest neighbors to SMOTE the training set.

   Step 2. $\mathbf{x} = (\mathbf{x}; \mathbf{1})$: Add intercepts to $\mathbf{x}$.

   Step 3. Initialize weights to be the zero-matrix.

   Step 4. Given the initial weights, learning-rate and maximum-iterations, use **Gradient Descent** to minimize the Log-Likelihood (given in the TextBook), and then obtains a trained weight matrix.

(3) **SMOTE**:

Input: feature matrix $\mathbf{x}$, label vector $\mathbf{y}$ (0-or-1 value) of training set, and $\mathbf{k}$ used for the number of nearest neighbors in the **KNN** algorithm.

Output: SMOTEd feature matrix and label vector.

The detailed inpelmentation could be found in original SMOTE paper. However, the original code takes in $\mathbf{T}, \mathbf{N}, \mathbf{k}$ as input, where $\mathbf{T}$ is the number of minority class, $\mathbf{N}$ is the amount of SMOTE, and $\mathbf{k}$ shares the same meaning as above. We could easily compute the value of $\mathbf{T}, \mathbf{N}$ using dataset $\mathbf{x}, \mathbf{y}$.

3. Implementation and Performance

In the assigned dataset, there are 5 classes for the samples. Therefore, there are 5 binary-logistic-regression classifiers in total.

Some parameters I use for all 5 classifiers:

learning-rate $= 0.0005$, **maximum iterations** $= 10000$,

$\mathbf{k} = 5$ (for **k**-nearest-neighbors).

The performance summary is given in Table (3).

表 2: Performance on test set: (1) recall, precision and error in each class (2) accuracy on the whole test test

Label	1	2	3	4	5	Accuracy
Recall	0.9274	0.9459	1.0000	1.0000 5	0.8333	
Precision	0.9824	0.7609	0.2143	0.8000	0.1818	0.9201
Error	0.0109	0.0058	0.0061	0.0020	0.0109	