学习小组 2 机器学习第二次作业

王锦宏、吴泽辉、陶少聪、王海东、叶泽林、陈耀顺

（中山大学智能工程学院，广东 深圳，518107）

摘要：基于信用卡用户的年收入(income)、信用卡每月账单(balance)、是否是学生(student)，预测用户是否会拖欠信用卡还款(default)。对数据集使用 10 折交叉验证，训练一个线性分类模型，预测信用卡用户是否会拖欠还款。

关键词：线性分类 (logistic regression)

1 描述 10 折交叉验证对数据集的处理

所谓交叉验证法，就是将一个数据集分为 K 份，然后取其中一份作为测试集，剩余 K-1 份作为训练集。然后，取另一份作为测试集，其余 K-1 份作为训练集。如此循环，直到每一份都做过测试集为止。本题当中采取的是 10 折交叉验证，即取 k=10 时对数据进行处理。每次试验都会得出相应的正确率（或差错率）。10 次的结果的正确率（或差错率）的平均值作为对算法精度的估计。一般还需要进行多次 10 折交叉验证（例如 10 次 10 折交叉验证），并且采取不同的划分数据集的方式，减小因样本划分不同而产生的误差。再求其均值，作为对算法准确性的估计。同时十折交叉验证法作为一种对数据处理结果的评估算法，也可以用于对不同模型拟合结果的一种选择。

2 描述所使用的线性模型

所用的模型：对数几率回归

$$ln\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}=\boldsymbol{\omega}^T\boldsymbol{x}+b$$

$$p(y=1|\boldsymbol{x})=\frac{e^{\boldsymbol{\omega}^T\boldsymbol{x}+b}}{1+e^{\boldsymbol{\omega}^T\boldsymbol{x}+b}}$$

$$p(y=0|\boldsymbol{x})=\frac{1}{1+e^{\boldsymbol{\omega}^T\boldsymbol{x}+b}}$$

对于输入的 $\mathbf{x}$，计算$\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})}$与设定的阈值比较，大于阈值的为正类，反之为反类

3 描述训练模型所使用的算法

为了便于表示和计算，令$\boldsymbol{\beta}=(\boldsymbol{\omega};b)$,$\hat{\boldsymbol{x}}=(\boldsymbol{x};1)$，$p_i(\boldsymbol{\beta})=p(y=i|\hat{\boldsymbol{x}};\boldsymbol{\beta})$，m 为样本数，利用最大化对数似然估计：

$$l(\boldsymbol{\beta})=\sum_{i=1}^{m}ln\,p\,(y_i||\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})$$

等价于最小化：

$$l(\boldsymbol{\beta})=\sum_{i=1}^{m}(-y_i\boldsymbol{\beta}^T\hat{\boldsymbol{x}}_i+ln(1+e^{\boldsymbol{\beta}^T\hat{x}_i}))$$

故所要求的参数向量为：

$$\boldsymbol{\beta}^*=\underset{\boldsymbol{\beta}}{argmin}l(\boldsymbol{\beta})$$

利用梯度下降算法得到最优解：

（1）取初值

$\boldsymbol{\beta}^{(0)}$=[0.1;0.1;0.1;0.1;0.1]，置 k=0;

（2）计算

$$l(\boldsymbol{\beta}^{(k)})=\sum_{i=1}^{m}(-y_i{\boldsymbol{\beta}^{(k)}}^T\hat{\boldsymbol{x}}_i+ln(1+e^{\boldsymbol{\beta}^{(k)T}\hat{x}_i}))$$

（3）计算梯度

$$\frac{\partial l(\boldsymbol{\beta}^{(k)})}{\partial\boldsymbol{\beta}^{(k)}}=-\sum_{i=1}^{m}\hat{\boldsymbol{x}}_i(y_i-p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta}^{(k)}))$$

并且一维搜索确定η_k得到

$$\underset{\eta_k\geq0}{minl}(\boldsymbol{\beta}^{(k)}-\eta_k\times(\frac{\partial l(\boldsymbol{\beta}^{(k)})}{\partial\boldsymbol{\beta}^{(k)}}))$$

（4）$\beta^{(k+1)}=\beta^{(k)}-\eta_k\times(\frac{\partial l(\beta^{(k)})}{\partial\beta^{(k)}})$

并计算$l(\boldsymbol{\beta}^{(k+1)})$,

当$||l(\boldsymbol{\beta}^{(k+1)})-l(\boldsymbol{\beta}^{(k)})||<\varepsilon$时

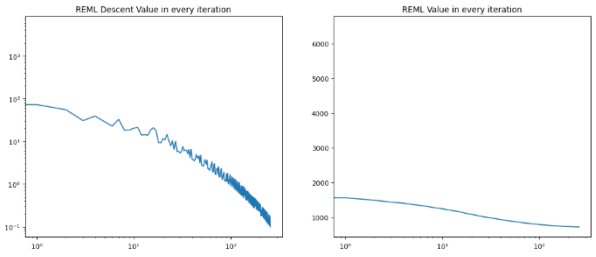
令$\boldsymbol{\beta}^*=\boldsymbol{\beta}^{(k)}$，停止迭代

（5）否则置 k=k+1,返回（2）

4 模型训练结果，训练集错误率和测试集错误率分析

训练过程的总结

我组提交的 T2.ipynb 在进行拟合的时候，同样可以对模型的训练情况进行画图。由于篇幅关系，全部的训练过程图此处不附。取使用源数据训练模型，并用模型验证源数据的情况：



在 T2 模型的训练中，全部的训练组普遍出现了梯度下降值出现了周期性震荡的情况，且随着训练集的不同，测试集震荡的幅度均不同。

十折中所有训练组平均 265 步左右即可收敛，

由最终求出的 ω 算出的正例与实际上的正例比较得出真正例、假正例、真反例、假反例的值。查准率为$\frac{\text{真正例}}{\text{真正例+假正例}}$，查全率为$\frac{\text{真正例}}{\text{真正例+假反例}}$，准确率为$\frac{\text{真正例+真反例}}{\text{正例+反例}}$。

我组的数据样本中，十折交叉验证的**平均查全率：0.381105，平均查准率：0.641764，平均准确率 0.971333**

从下面的图 2 也可看出训练集和测试集之间这三种率很接近，因此泛化效果比较好。

训练组	拟合的ω				
1	[4.778	−0.160	−3.414	−2.600	−6.023]
2	[4.724	−0.216	−3.375	−2.494	−5.879]
3	[4.717	−0.161	−3.404	−2.568	−5.982]
4	[4.768	−0.204	−3.425	−2.486	−5.921]
5	[4.667	−0.190	−3.416	−2.430	−5.855]
6	[4.722	−0.166	−3.417	−2.540	−5.966]
7	[4.763	−0.186	−3.400	−2.530	−5.941]
8	[4.779	−0.187	−3.471	−2.509	−5.990]
9	[4.712	−0.161	−3.389	−2.559	−5.957]
10	[4.817	−0.203	−3.400	−2.569	−5.979]

图 1 十折验证训练模型分别拟合得到的 ω

查准率	查全率	准确率	查准率	查全率	准确率
0.631579	0.352941	0.967778	0.631579	0.352941	0.967778
0.640000	0.400000	0.963333	0.638158	0.363296	0.972222
0.625000	0.357143	0.973333	0.625000	0.357143	0.973333
0.692308	0.321429	0.974444	0.692308	0.321429	0.974444
0.565217	0.419355	0.968889	0.565217	0.419355	0.968889
0.650000	0.382353	0.968889	0.650000	0.382353	0.968889
0.705882	0.480000	0.980000	0.705882	0.480000	0.980000
0.590909	0.406250	0.968889	0.590909	0.406250	0.968889
0.846154	0.343750	0.974444	0.655172	0.401408	0.971605
0.470588	0.347826	0.973333	0.655172	0.401408	0.971605
			0.635870	0.381107	0.971444

图 2 十折交叉验证对自身、对测试样本的拟合结果
最后一行为全数据样本训练后对自身的拟合结果

5 模型训练过程中的收获

5.1 基础知识

加深了对 python 的熟悉，如学习了 dataframe 等机器学习常用的数据储存方式，加强了 pandas、numpy、matplotlib 等库的使用；回顾了课本上的数学公式；将算法用代码实现，并用 python 完成矩阵运算，提升了编程能力；任务管理上，训练了 JupyterNotebook 集成环境的使用、利用 Github 完成了小组合作。

5.2 数据处理

学习了数据可视化时，对多变量情况需要进行归一化，所得图片才有对比价值；面对类别样本数不平衡问题，使用阈值移动处理，模型能有更好结果。

5.3 模型调试

学习步长的选取，太大会无法收敛，太小会使训练速度过慢；调试中建议选取较小的学习步长，虽然学习时间较长，但能避免不收敛的情况。

5.4 代码改进

使用矩阵运算，相比循环运算，能有效的提高效率；尝试了除具体迭代外，其他的判准，尝试了对模型效率的提高。