数据分析-分类预测

吴韬略*

中国科学技术大学 大数据学院，合肥

【摘　要】 本次实验旨在对之前的实验数据进行分类预测，实现了决策树，KNN 等算法，采用交叉验证的方法训练模型。

【关键词】 分类预测，决策树，KNN，K-FOLD，MLP，集成学习

1 数据预处理

数据预处理步骤同实验三，选取与 REPEAT 列相关系数较大的 11 个特征作为数据集：

$$[ISCEDL\ ST005Q01TA\ ST013Q01TA\ OCOD2\ ST007Q01TA$$

$$OCOD3\ OCOD1\ HISCED\_D\ MISCED\_D\ IMMIG\ ST019BQ01T]$$

2 算法流程

$Step1$ **创建模型**

$Step2$ **随机划分数据集**

$Step3$ **交叉验证训练**

$Step4$ **保存模型并输出预测性能**

3 关键技术

(1) **K-FOLD 交叉验证：** K-fold 交叉验证是一种在机器学习中常用的评估模型性能的方法。它将数据分成 k 份，每次将一份作为验证集，其余 k-1 份作为训练集，进行 k 次训练和验证，最后将 k 次的结果平均，得到最终的模型性能。

(2) **决策树：** 决策树是一种以树形数据结构来展示决策规则和分类结果的模型，作为一种归纳学习算法，其重点是将看似无序、杂乱的已知数据，通过某种技术手段将它们转化成可以预测未知数据的树状模型，每一条从根结点（对最终分类结果贡献最大的属性）到叶子结点（最终分类结果）的路径都代表一条决策的规则。决策树的生成算法有 ID3、C4.5 和 C5.0 等。

(3) **KNN：**KNN 代表 K 近邻。这是一种用于分类和回归的简单机器学习算法。该算法的工作原理是在训练集中找到与输入数据点最近的 k 个数据点，然后使用这些邻居进行预测。

4 算法性能及调参

4.1 决策树

初始时不调参，结果如图1(a)。存在严重的过拟合，下面一步步调参：

$Step1$ **高度** h

经过反复调整，限制树高度为 8 时过拟合现象消失，结果如图1(b)。

$Step2$ **最小分裂样本数** n

每个结点只有大于 n 才会分裂，可以防止决策树过高。取值 100，结果如图1(c)。

$Step3$ **最小信息增益量**

调整之后好像没有什么变化，取值很小，和取 0 效果没有区别。结果如图1(d)。

$Step4$ **最大复杂度**

还是没有变化取值接近 0。结果如图1(e)。

4.2 KNN

初始时不调参，结果如图2(a)。也存在较严重的过拟合，下面一步步调参：

$Step1$ **邻居数量**

经过反复调整，邻居数量为 20 时过拟合现象消失，结果如图2(b)。

$Step2$ **权重**

取权重为距离的倒数，训练集准确率大大提升，终于破 9，可是却再次出现过拟合。结果如图2(c)。

$Step3$ **叶子大小**

传递给 BallTree 或者 KDTree[1] 函数的叶子大小参数，取值 20。结果如图2(d)。

$Step4$ **距离度量**

选取的距离度量方式，选择余弦距离。结果如图2(e)。

注：由于 KNN 算法运行较慢，本节采用多线程实现。

4.3 MLP

出于好奇心，本实验尝试了 MLP，与传统的非神经网络方法做对比。初始时不调参，结果如图3(a)。下面一步步调参：

$Step1$ **最大迭代次数**

由于后续调参经常无法在短时间收敛，故更改最大迭代次数为 1000。

$Step2$ **隐含层神经元个数**

[1] KNN 中寻找最近邻居的算法

	test	train
accuracy	0.779821	0.913911
f1-score	0.769146	0.908541

(a) 未调参

	test	train
accuracy	0.818417	0.828102
f1-score	0.797046	0.807865

(b) 高度 8

	test	train
accuracy	0.819058	0.824842
f1-score	0.797214	0.803664

(c) 最小分裂样本数 100

	test	train
accuracy	0.819058	0.824842
f1-score	0.797214	0.803664

(d) 最小信息增益量 0

	test	train
accuracy	0.817895	0.824159
f1-score	0.795237	0.802160

(e) 最大复杂度 8e-5

图 1: 决策树结果

	test	train
accuracy	0.795592	0.846468
f1-score	0.783385	0.837385

(a) 未调参

	test	train
accuracy	0.817918	0.829687
f1-score	0.792380	0.806172

(b) 邻居数量 20

	test	train
accuracy	0.800841	0.913811
f1-score	0.782646	0.908550

(c) 距离倒数权重

	test	train
accuracy	0.809439	0.837638
f1-score	0.782636	0.815596

(d) 叶子大小 20

	test	train
accuracy	0.808940	0.838078
f1-score	0.782294	0.816337

(e) 余弦距离度量

图 2: KNN 结果

使用三分法[4]编程求解最佳神经元个数，取值为 90，结果如图3(b)，因为不调参时默认取值为 100，相差较小，故得分提升不大。

$Step3$ **隐含层层数**

取值 2 层，结果如图3(c)

$Step4$ **学习率更新方式**

取值 adaptive：只要训练损失不断减少，'adaptive' 就会使学习率保持在'learning_rate_init' 的水平。每当连续两个历时未能使训练损失至少减少 tol，或者在'early_stopping' 开启的情况下未能使验证分数至少增加 tol 时，当前的学习率就被除以 5。结果如图3(d)。

$Step5$ **热重启**

每一次迭代时使用上一次求解得到的最优解作为初始值。结果如图3(e)。

	test	train
accuracy	0.821196	0.830038
f1-score	0.801858	0.812139

(a) 未调参

	test	train
accuracy	0.821030	0.830216
f1-score	0.802127	0.812751

(b) 神经元个数 90

	test	train
accuracy	0.819700	0.830614
f1-score	0.802631	0.814356

(c) 隐含层 2 层

	test	train
accuracy	0.823049	0.829153
f1-score	0.804488	0.811563

(d) adaptive 更新

	test	train
accuracy	0.821386	0.828613
f1-score	0.803093	0.811230

(e) 热重启

图 3: MLP 结果

4.4 组合特征

使用实验三中的自构建特征，并去除违规特征进行尝试，模型选择为效果最好的 MLP 模型。结果如图4。

4.5 集成学习

将上述三个模型进行集成学习，结果如图5(a)。调参'voting' 投票方式，由'hard' 改为'soft'，即基于预测概率之和的 argmax 来预测类标签，结果如图5(b)。

注：由于集成学习算法运行较慢，本节采用多线程实现。

	test	train
accuracy	0.815068	0.815080
f1-score	0.787832	0.787846

图 4: 自构建特征

	test	train
accuracy	0.818987	0.832021
f1-score	0.797095	0.811996

(a) hard

	test	train
accuracy	0.819438	0.836534
f1-score	0.800828	0.820071

(b) soft

图 5: 集成学习结果

5 总结

即使使用了三个模型并且多次调参，最后甚至集成了三者学习，得分最高还是不到 0.83，提升的幅度并不大，失败原因可能出在实验三的数据预处理上和调参方法上（比如神经网络初值可以先通过遗传算法确定等）。经过本次实验，我深刻体会到建立模型并且调参需要扎实的理论基础，明白每一个参数可能的影响，如此才能达到目标。

参考文献

[1] Christian Borgelt. An implementation of the fp-growth algorithm. In *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, pages 1–5, 2005.

[2] Liu Qi. Prof. qiliu’s slides on ad2023. `http://staff.ustc.edu.cn/~qiliuql/files/AD2023/4.1.pdf`.

[3] Liu Qi. Prof. qiliu’s slides on ad2023. `http://staff.ustc.edu.cn/~qiliuql/files/AD2023/4.2.pdf`.

[4] 王嵘冰, 徐红艳, 李波, and 冯勇. Bp 神经网络隐含层节点数确定方法研究. 计算机技术与发展, 28(4):31–35, 2018.