中山大学数据科学与计算机学院

移动信息工程专业-人工智能

本科生项目报告

（2017-2018 学年秋季学期）

课程名称：**Artificial Intelligence**

教学班级	M1	专业（方向）	移动互联网
组号	22	组名	猪猪侠
学号	15352102	姓名	韩硕轩

一、 Project 最终结果展示

1. 最终结果

写报告时（1 月 9 日）的结果和排名：

22	39	来自底层的仰望	0.909771	0.619693	68.38214	20	13	35	21
23	13	渣渣求及格队	0.90872	0.647153	74.36153	24	3	40	22
24	22	猪猪侠	0.909895	0.590824	60.74683	19	32	19	23
25	5	给章鱼征婚	0.90872	0.600366	59.33084	24	28	17	24
26	11	佩服三连	0.90872	0.604125	61.31265	24	25	22	25

2. 组内分工

戴斯铭：二元分类

陈德和：多元分类

韩硕轩（笔者）：回归

3. 个人工作

时间	工作
12.23 到 1.8	处理回归数据集（多个阶段）；
12.24	尝试 kNN，跑回归结果；
12.25 到 12.27	学习 SVR；
12.27	调 SVR 库跑结果，然后放弃实现……
12.28 到 1.8	调试 BPNN，跑回归结果；
1.2	做展示 ppt；
1.9	写最终报告；

二、 工作流程

1. 算法简介

1. 使用过的算法和效果

算法	参数	效果（回归测试集 rmse）
SVR（调库）	rbf 内核	112.3
KNN	K=30；onehot 编码；欧氏距离	109.2
BPNN	学习率 0.01；两个隐藏层 80/20 个节点；输出层 1 个节点	60.7

2. 采用的辅助手段

①改用 onehot 编码

将一些离散的属性变为 onehot 编码，比如季节原来是用 1234 区分，现在 1 就是 0001，4 就是 1000 这样。因为 onehot 编码的优点是可以扩充属性，原先是 1 个现在是多个，而且属性不同值之间的距离变得合理，比如原先季节 1 和 2 的距离比 1 和 4 的距离更近，而改用 onehot 之后它们就是一样的了，与实际更相符。

改用 onehot 之后我的 BPNN 从一开始的 130 左右降到了 70，效果明显。

部分代码展示：

```
if is_equal(temp[4], 1):
    temp = np.hstack((temp, [1, 0, 0, 0]))
elif is_equal(temp[4], 2):
    temp = np.hstack((temp, [0, 1, 0, 0]))
elif is_equal(temp[4], 3):
    temp = np.hstack((temp, [0, 0, 1, 0]))
else:
    temp = np.hstack((temp, [0, 0, 0, 1]))
temp = np.delete(temp, 4, axis=0) #delete 'season' and append onehot
```

②相关性分析

改成 onehot 之后，我又尝试把每一列与 cnt 列算一遍相关系数，然后将相关系数绝对值小于 0.1 的列都删除。这样跑出来的效果最好，也就是我最终提交的 60.7。

相关系数可以表示两个变量间的线性关系，取值在-1 到 1 之间[2]。取-1 的时候表示两个变量完全负相关，1 表示完全正相关，而 0 则表示不相关。我用 numpy 库里的 corrcoef 方法可以得到这个系数矩阵，取出它的[0][1]即为相关系数。

部分代码展示：

```
dlt = []
for i in range(0, trainMat.shape[1]):
    attr = trainMat[:, i].T.tolist()[0]
    cor = corrcoef(attr, label.T.tolist()[0])[0][1]
    print cor
    if math.fabs(cor) < 0.01:
        dlt.append(i)
```

3. BPNN 简介

反向传播神经网络是我们之前实验中使用过的，这一次拿来作为主要的算法。由于它能够处理高度非线性的数据集[3]，所以适应性比较强，在本次实验中会有比较好的效果。

主要的过程是先正向传播，从输入层到隐藏层再到输出层，计算出结果和误差。然后再将误差反向传播，更新隐藏层的权重。通过不断迭代，达到一个比较好的效果。

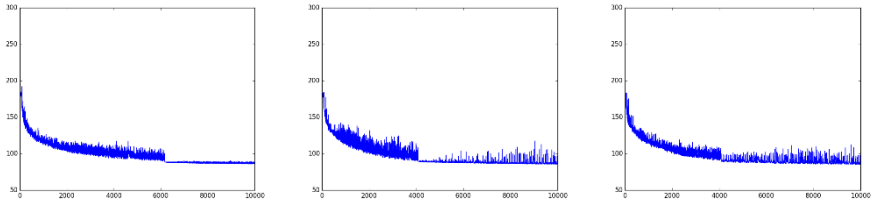
但是 BPNN 的参数调整很难，不小心就会导致预测值全都一样或者过拟合，所以需要细致地调整参数和处理数据集，否则效果不如 kNN。调参和数据集处理我在其他部分有说明。

2. 调参过程

表现最好的算法是 BPNN。我的调参过程是伴随着数据集处理的。然而每次对于不同的数据集，我把参数调到接近最优的过程是相似的。

BPNN 里面要调的参数有学习率、隐藏层数、每层节点数和迭代次

数。下图是我完全处理完数据集之前跑的几张图。我控制其他变量不变，调整学习率为不同的值，配合画图跑出来的效果如下：

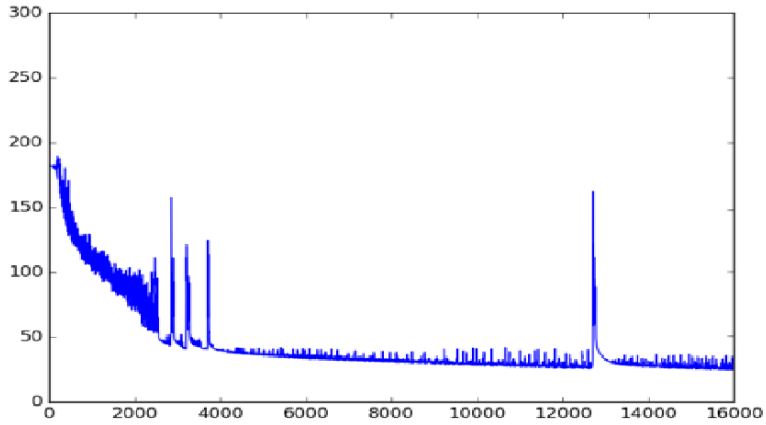


学习率： 0.01 0.012 0.016

从左到右依次是学习率为 0.01,0.012 和 0.016 的效果，可以看出 0.01 在迭代后期的稳定性比较好，而后两者震荡比较大，容易迭代完之后得出较差结果。而如果小于 0.01，收敛速度又太慢。所以我让初始学习率为 0.01，每当 rmse 下降 10，就减小一点学习率，于是曲线变得稳定了很多。

再比如迭代次数，我一开始都是迭代好几万次，最终训练集的 rmse 可以一直降到 23，然而提交上去发现结果高达八十多。分析原因应该是产生了过拟合，所以我减少迭代次数，多次提交得出下表：

迭代次数	训练集 rmse	测试集 rmse
1000	48.1	64.4
1500	45.3	60.7
2000	42	62.2
3000	37.6	70.8
5000	31.3	77.6
10000	25.5	90.2
15000	23.9	95



所以最终决定将迭代次数控制在 1500 次。

3. 数据集分析

①二元分类数据集

二元数据集属性列很整齐，容易上手，每行前面是属性，最后一列是标签。Type 属性是字符串需要转化成数字。后面的列里面的数字跨度有大有小，需要预处理加上归一化，否则跨度大的属性将会掩盖跨度小的属性的作用。

然而队友经过尝试发现不处理用 KNN 反而效果更好。我们讨论以后觉得可能是因为跨度大的属性本身就比较重要，这样一来正好就起到了决定性作用，而跨度小的属性几乎不起作用。另外经过队友尝试，发现无论单层还是多层决策树效果都没有 KNN 好，可见该数据集线性可分性比较差。

②三元分类数据集

三元分来数据集是最大最难处理的，首先是 LOW、MID、HIGH 的标签，然后是一段文本。

这段文本里面包含一些在情感分类中不起作用的词，因此数据预处

理需要削弱它们的影响。此外，利用之前用过的编码方式很可能会爆内存跑不起来，所以降维处理数据集很有必要。

③回归数据集

回归数据集相对前面两个来说较小，属性列比较整齐，实际意义就是租单车情况的统计，很好理解。

训练集中存在一些缺失值，处理方法可以是直接舍弃，不过我们组选择将它处理成当月的均值。另外有一些属性全 0 的数据，那是每个月的租车总和，可以直接舍弃。除去这两者，数据集就是有效的 16637 个训练文本和 742 个测试文本。原始的文本在 BPNN 下的效果并不好，甚至一直会预测出全部一样的值，这需要另外处理数据。我先是增加了属性列，由于 BPNN 的实验用的是相似的数据集，我就按照上一次的 readme 将这次的同样处理，根据日期算出了星期、季节等属性，还去原网站[1]上获取了 2011 到 2012 年间的所有假日添加到属性中，除了临时用户和注册用户都加上去了。其余的处理包括处理成 onehot 编码和相关性分析，这在【工作流程】的算法分析之“采用的辅助手段”里面有写。

4. 集成学习方法(AdaBoost)

本次我负责的回归任务没有用到 AdaBoost，另外两个任务队友有用到。Adaboost 会针对一个训练集训练多个分类器，不同于 Bagging，它是按顺序进行的。所有选出的训练样本都会有一定的几率参与下一次学习。如果它被正确分类，那么它下一次参与学习的几率就会降低，反之升高。也就是说我们更关注错误的样本。

二元分类戴斯铭先后尝试了单层二叉决策树，单层多叉决策树和多层决策树，三元分类陈德和尝试了 adaboost 与 OVO 模型和 OVA 模型的结合，同样在单层二叉决策树上做了尝试。

结果如下（队友供图）：

数据集	子模型	结果
二元分类	单层二叉决策树	0.7513
	单层多叉 CART 决策树	0.7376
	多层决策树	0.74
三元分类	单层二叉决策树	55.2124

跑出来的结果二元上 adaboost 远没有 knn 效果好，原因可能并不是参数调整得不好，事实上我了解到队友在这里花了不少时间。分析数据集之后应该这个是数据集的线性可分性比较差，使得 knn 能够训练出更好的模型。

三、 引用

[1]. http://dchr.dc.gov/page/holiday-schedule

[2]. https://baike.baidu.com/item/%E7%9B%B8%E5%85%B3%E7%B3%BB%E6%95%B0/3109424?fr=aladdin

[3]. https://baike.baidu.com/item/%E5%8F%8D%E5%90%91%E4%BC%A0%E6%92%AD%E7%AE%97%E6%B3%95/522948?fr=aladdin

四、 课程总结

这门课的理论课任务很轻松，不过理论本身还是很需要数学功底。老师上课讲得很细致，能让我们理解得比较透彻。如果我没选而是自学肯定效果没这么好。

实验课任务相对比较繁重，不过也许是上过操作系统实验的原因，我对这学期任务量和 TA 风格都比较适应，我知道这都是对我们有好处的。

从内容上看，我们学了 KNN、NB、PLA、LR、DT、BPNN 这些在机器学习领域比较基础的算法。理论课就会讲得很深入很完整，把整个算法的原理都分析得很清楚。然后实验课就会介绍实践的方法，自己实现一遍就觉得印象更深刻。

做实验的过程总体还是很有趣的。人工智能和操作系统相比写报告时间短了，打代码时间长了。写代码的过程是对代码能力的一种考验，特别是改用 python 之后我发现代码变得更灵活，所以需要更加严谨地使用否则会发生意想不到的错误。遇到问题的时候就经常会用理论课的原理去分析，然后对于这个理论的理解就会加深。

不管以后如果继续从事人工智能方面的研究，也应该多是调库实现。所以很珍惜这个学期自己实现这么多算法的过程，也很感谢老师和 TA 辛勤的付出，让我收获很多。