

基于深度学习的诊断心脑血管疾病分类方法

黄璐,毛晓艳

(北京控制工程研究所,北京 100094)

摘要:在人工智能与机器学习飞速发展的大环境下,考虑可以将其应用于医学研究领域和医疗行业。在诊断心脑血管疾病时,可以使用深度学习方法,通过大数据挖掘协助医生进行分析判断与病情诊断。文中采用支持向量机 SVM,集成学习算法 XGBoost 和 BP 神经网络三种方法进行对比,调节神经网络参数,训练分类器。寻找最优的分类方式,使其达到最高的准确率,协助医生对病人进行心脑血管疾病的判断。经数字仿真与模拟实验,进行数据挖掘与样本特征分析,调节参数,绘制散点图和特征相关性热力图分析,发现通过神经网络的方法训练分类器可以以较高的准确率应用于医学领域中。

关键词:神经网络;分类器;深度学习;特征分析

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2021)0036-05

Classification Method of Cardiovascular and Cerebrovascular Diseases Based on Deep Learning

HUANG Lu, MAO Xiao-yan

(Beijing Institute of Control Engineering, Beijing 100094, China)

Abstract: In the context of the rapid development of artificial intelligence and machine learning, it can be considered to be applied to medical research and medical industry. In the diagnosis of cardiovascular and cerebrovascular diseases, deep learning can be used to assist doctors in analysis and diagnosis through big data mining. Three methods of support vector machine (SVM), integrated learning algorithm XGBoost and BP neural network are used to compare, adjust the neural network parameters and train the classifier. To find the best classification method, so as to achieve the highest accuracy, to assist doctors in the diagnosis of patients with cardiovascular and cerebrovascular diseases. Through digital simulation and simulation experiments, data mining, sample feature analysis, parameter adjustment, scatter diagram drawing and feature correlation heat map analysis are carried out. It is found that the neural network training method can be applied to medical field with high accuracy.

Key words: neural network; classifier; deep learning; characteristics analysis

0 引言

心脑血管疾病是心脏血管和脑血管疾病的统称,泛指由于高脂血症、血液黏稠、动脉粥样硬化、高血压等所导致的心脏、大脑及全身组织发生的缺血性或出血性疾病。其病因有很多方面,经医学研究表明,其主要有以下几个方面:高血压、血液粘稠、吸烟、酗酒、糖尿病,其他如肥胖、年龄、性别等。心血管疾病的检查需要医生根据患者的各项指标来综合评定。为此,文中设计一种模型,根据患者的基本信息及相关检查结果来辅助医生诊断。此次诊断疾病任务是一个多分类问题。分类是机器学习的基础问题之一,考虑到这个问题的复杂性适中,数据量中等,采用多种机器学习算法与深度学习算法对其进行分析。

1 简介

收集了大约 70 000 个患者的基本信息,一共 12 个特征,其中包含训练集和测试集的比例大约为 7 : 3,即约 50 000 条数据以供训练。分为训练集、测试集、验证集。将预测结果和验证集结果比较得出的准确率作为模型评价标准。

2 理论与方法

此问题可以归类为二分类问题。算法选择分为 3 部分,特征提取、分类器、神经网络。考虑各个算法存在的问题及表现,此次实验使用的是 SVM,集成学习算法 XGBoost 和 BP 神经网络。

支持向量机(support vector machine, SVM)是一

种二分类模型,它的基本模型是定义在特征空间上的间隔最大的线性分类器,间隔最大使它有别于感知机;SVM 还包括核技巧,这使它成为实质上的非线性分类器^[1]。SVM 的学习策略就是间隔最大化,可形式化为一个求解凸二次规划的问题,也等价于正则化的合页损失函数的最小化问题^[2]。SVM 的学习算法就是求解凸二次规划的最优化算法^[3]。SVM 学习的基本思想是求解能够正确划分训练数据集并且几何间隔最大的分离超平面^[4]。对于线性可分的数据集来说,这样的超平面有无穷多个(即感知机),但是几何间隔最大的分离超平面却是唯一的^[5]。对于输入空间中的非线性分类问题,可以通过非线性变换将它转化为某个维特征空间中的线性分类问题,在高维特征空间中学习线性支持向量机^[5]。

XGBoost 模型是大规模并行 boosting tree 的工具,它是目前较好的开源 boosting tree 工具包。在了解 XGBoost 算法基本原理之前,需要首先了解 Boosting Tree 算法基本原理。Boosting 方法是一类应用广泛且非常有效的统计学习方法。它基于这样一种思想:对于一个复杂任务来说,将多个专家的判断进行适当的综合所得出的判断,要比任何一个专家单独的判断要好^[6]。XGBoost 有两大类接口:XGBoost 原生接口和

scikit-learn 接口,而且 XGBoost 可以完成分类和回归两种任务。此次实验使用的是 XGBoost 基于 scikit-learn 接口的分类。

任务中使用神经网络时,大多数是使用 BP 算法进行训练^[7]。BP 神经网络就是一个“万能的模型+误差修正函数”,每次根据训练得到的结果与预想结果进行误差分析,进而修改权值和阈值,一步一步得到能输出和预想结果一致的模型^[8]。BP 网络由输入层、隐藏层、输出层组成^[9]。

3 实验仿真结果及分析

3.1 分类器为 SVM

对于模型,主要考虑了如下几个常见参数:惩罚系数 C、核函数 kernel、n 折交叉验证、特征数量 k。并选择出最优参数。在数据上,采取了两种特征选择方法:特征相关性分析并手动剔除相关性低的特征、指标筛选 K 个最优特征。考虑模型集成,使用 bagging 的方法。如表 1 所示:(1)改变惩罚因子 C;(2)改变核 kernel;(3)改变 n 折;(4)改变特征数量。前三个都建立在 SelectKBest(f_classif, k=10),指标筛选 10 个最优特征。

表 1 SVM 调参准确率

模型参数	准确率/%	n 折交叉验证	训练时间/s
C=0.2, kernel=linear	72.65	7 : 3	38.43
C=0.6, kernel=linear	72.71	7 : 3	43.66
C=1.5, kernel=linear	72.75	7 : 3	56.00
C=0.6, kernel=linear+bagging	72.66	7 : 3	457.02
C=1.5, kernel=poly	66.17	7 : 3	56.38
C=1.5, kernel=rbf	72.68	7 : 3	57.80
C=1.5, kernel=sigmoid	56.51	7 : 3	64.51
C=0.6, kernel=linear	72.59	8 : 2	59.47
C=0.6, kernel=linear	72.56	6 : 4	33.04
C=0.6, kernel=linear(手动剔除相关性较低的特征)	72.16	7 : 3	48.31
C=0.6, kernel=linear(不筛选任何特征)	72.66	7 : 3	47.93

根据与标签的相关性分析,绘制表格如图 1 所示。

id	1												
age	0.0031	1											
gender	-0.0076	-0.019	1										
height	-0.0055	-0.08	0.5	1									
weight	0.00073	0.054	0.15	0.28	1								
ap_hi	0.0072	0.019	0.006	0.0089	0.029	1							
ap_lo	-0.0018	0.017	0.016	0.0091	0.043	0.015	1						
cholesterol	0.0011	0.15	-0.037	-0.054	0.14	0.021	0.021	1					
gluc	0.0041	0.096	-0.02	-0.023	0.1	0.011	0.012	0.45	1				
smoke	-0.0077	-0.045	0.34	0.19	0.065	-0.0015	0.0054	0.0098	-0.0018	1			
alco	-0.0034	-0.03	0.17	0.096	0.068	0.001	0.011	0.033	0.01	0.34	1		
active	0.0023	-0.01	0.0067	-0.0084	-0.018	0.0022	0.0036	0.012	-0.0085	0.031	0.03	1	
cardio	0.0039	0.24	0.0039	-0.014	0.18	0.056	0.062	0.22	0.088	-0.019	-0.0097	-0.038	1

图 1 特征相关性分析

分析数据可知,该问题为二分类问题。在五万条数据中,两类数据分别为 25 048 条和 24 952 条,约为 1 : 1。其中,分析身高和体重数据发现,有些数据为异常数据,例如 height 为 178cm,而 weight 为 11 kg 等很明显不是正常的身高体重数据。因此去掉异常数据再进行训练,但预测准确率很低,可能是这些数据对训练模型影响较大,不能通过简单地判断来随意剔除。

首先考虑到有 11 个特征来判断是否患心血管疾病,不筛选特征,控制变量惩罚系数 C 、 n 折交叉验证和 SVM 的核 kernel,测试准确率 72.66%;进行特征与标签的相关性分析,绘制表格得到相关性绝对值排序,手动剔除相关性较低的两个特征(分别是 id 和 alco),得到测试准确率 72.16%,准确率反而降低,分析原因可能是手动筛选特征并不够准确,并且筛选掉两个特征可能对准确度负影响大;因此使用 SelectKBest(f_{classif} , $k=10$),由电脑自动筛选出最优的十个特征进行模型训练,控制变量得到预测准确率 72.71%,准确率提高,但效果并不明显。在此基础上调整惩罚函数 C 、 n 折交叉验证和 SVM 的核 kernel,测试是否能提高准确率。首先调整 C , C 越高,越容易出现过拟合, C 越小,越容易出现欠拟合,无论过大或过小,模型泛化能力变差。随着 C 增大,预测准确率提高,但当 C 再继续增大至超过 1.5 时,准确率几乎无提升。取 $C=1.5$,模型准确率达到 72.75%;考虑改变 n 折交叉验证,调整训练集与测试集之间的比值,分别取 7 : 3,

6 : 4,8 : 2,发现在取 7 : 3 时预测准确率最好,为 72.71%;最后改变 kernel,分别取 linear、poly、rbf、sigmoid、linear 和 bagging 相结合这几种核,发现取线性核(linear)时预测准确率最好。综上,经验证可得,取 $C=1.5$,kernel=linear,训练集:测试集=7 : 3 时预测准确率最高,为 72.75%。

3.2 分类器为 XGBoost

3.2.1 数据分析和预处理

由图 2 可知,对该数据集特征分析如下。gender: 性别,1 表示女性,2 表示男性,均值约为 1.35 说明该数据集中女性偏多,查看男女人数,发现女性数据约为男性数据的两倍,但男女性的类别数据都是均衡的。height & weight:身高的最小值为 55 cm,最大值为 250 cm,体重的最小值为 14.325 kg,最大值为 200 kg,明显都是异常数据。画出 height 和 weight 的散点图,可以更直观地看出异常值。这里选择将身高低于 120 cm 或高于 200 cm 或者体重小于 25 kg 或大于 150 kg 的值都去除。

ap_hi & ap_lo:血压高压和血压低压,从样本数据中可以看到,高压的最大值为 16 020,最小值为-150,低压的最大值为 11 000,最小值为 0,明显是异常数据,去除血压小于 30 或大于 300 的数据。cardio:类别标签,均值接近 0.5,说明数据整体是均衡的。进行上述数据分析处理后,对数据进行归一化处理。然后将其分为训练集的输入与标签和测试集。

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 13 columns):
 id                50000 non-null int64
 age               50000 non-null int64
 gender            50000 non-null int64
 height            50000 non-null int64
 weight            50000 non-null float64
 ap_hi             50000 non-null int64
 ap_lo             50000 non-null int64
 cholesterol       50000 non-null int64
 gluc              50000 non-null int64
 smoke            50000 non-null int64
 alco              50000 non-null int64
 active            50000 non-null int64
 cardio            50000 non-null int64
dtypes: float64(1), int64(12)
memory usage: 5.0 MB
None
```

	id	age	gender	height	weight
count	50000.000000	50000.000000	50000.000000	50000.000000	50000.000000
mean	24999.500000	19464.791580	1.347020	164.366940	74.230850
std	14433.901067	2468.926565	0.476027	8.188912	14.325484
min	0.000000	10798.000000	1.000000	55.000000	11.000000
25%	12499.750000	17662.000000	1.000000	159.000000	65.000000
50%	24999.500000	19703.000000	1.000000	165.000000	72.000000
75%	37499.250000	21321.000000	2.000000	170.000000	82.000000
max	49999.000000	23690.000000	2.000000	250.000000	200.000000

	ap_hi	ap_lo	cholesterol	gluc	smoke
count	50000.000000	50000.000000	50000.000000	50000.000000	50000.000000
mean	128.744780	96.981460	1.363520	1.225940	0.088220
std	154.455954	200.208981	0.677187	0.572099	0.283617
min	-150.000000	0.000000	1.000000	1.000000	0.000000
25%	120.000000	80.000000	1.000000	1.000000	0.000000
50%	120.000000	80.000000	1.000000	1.000000	0.000000
75%	140.000000	90.000000	1.000000	1.000000	0.000000
max	16020.000000	11000.000000	3.000000	3.000000	1.000000

	alco	active	cardio
count	50000.000000	50000.000000	50000.000000
mean	0.053360	0.802580	0.499040
std	0.224753	0.398056	0.500004
min	0.000000	0.000000	0.000000
25%	0.000000	1.000000	0.000000
50%	0.000000	1.000000	0.000000
75%	0.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000

图 2 数据集基本信息

3.2.2 建立模型及调参

构建一个 XGBoost 分类器,给参数一个合适的初始值。

(1)首先确定学习速率和估计器数目。

选择较高的学习速率 0.1,然后在固定其他参数初始值的情况下使用 XGBoost 内置的 CV 函数选择对

应于此学习速率的理想决策树数量。经过验证测试,对应的最佳迭代次数为 150。

(2)使用 gridsearch 进行参数调优。

使用 gridsearch 进行参数的调优,因为参数组合过多,所以采用分步调优,部分调参结果展示如表 2 所示。

表 2 XGBoost 调参结果

模型参数	testscore	trainscore	训练时间/s
learning_rate:0.1,"max_depth":3,"min_child_weight":3	0.802 786	0.810 365	7.436 844
learning_rate:0.1,"max_depth":3,"min_child_weight":4	0.802 569	0.810 165	7.285 485
learning_rate:0.1,"max_depth":4,"min_child_weight":3	0.802 815	0.816 897	8.254 802
learning_rate:0.1,"max_depth":4,"min_child_weight":4	0.802 623	0.816 648	9.563 527
learning_rate:0.1,"max_depth":5,"min_child_weight":3	0.801 693	0.825 619	9.870 789
learning_rate:0.1,"max_depth":5,"min_child_weight":4	0.801 412	0.824 979	9.395 601
gamma:0.05,"learning_rate":0.1,"max_depth":4,"min_child_weight":3	0.802 712	0.816 912	12.775 502
gamma:0.1,"learning_rate":0.1,"max_depth":4,"min_child_weight":3	0.802 871	0.816 959	11.940 12
gamma:0.15,"learning_rate":0.1,"max_depth":4,"min_child_weight":3	0.802 681	0.817 035	11.325 077
gamma:0.1,"learning_rate":0.1,"max_depth":4,"min_child_weight":3,"subsample":0.7	0.802 09	0.816 808	10.663 644
gamma:0.1,"learning_rate":0.1,"max_depth":4,"min_child_weight":3,"subsample":0.8	0.802 871	0.816 959	9.734 894
gamma:0.1,"learning_rate":0.1,"max_depth":4,"min_child_weight":3,"subsample":0.9	0.802 338	0.816 917	10.209 504
colsample_bytree:0.7,"gamma":0.1,"learning_rate":0.1,"max_depth":4,"min_child_weight":3,"subsample":0.8	0.802 489	0.816 246	11.231 301
colsample_bytree:0.75,"gamma":0.1,"learning_rate":0.1,"max_depth":4,"min_child_weight":3,"subsample":0.8	0.802 871	0.816 959	10.656 795
colsample_bytree:0.8,"gamma":0.1,"learning_rate":0.1,"max_depth":4,"min_child_weight":3,"subsample":0.8	0.802 871	0.816 959	9.833 663

3.3 分类器为 BP 神经网络

本次实验中先测试剔除 id 特征的情况下模型的预测结果,控制隐藏层数量、每层的单元个数,迭代次数作为 BP 的调试结果如下:

第一层:(100, 200)

第二层:(100, 200)

第三层:(无,100, 200)

迭代次数:(500,1 000)

共进行 $2 \times 2 \times 3 \times 2 = 24$ 组实验。测试集和验证集

按 8:2 的比例随机划分,见表 3。

表 3 BP 神经网络调参准确率

第一层隐藏层	第二层隐藏层	第三层隐藏层	迭代次数	准确率	训练时间
100	100	无	500	0.672 1	222.38
100	100	无	1 000	0.670 9	253.33
100	100	100	500	0.694 2	92.39
100	100	100	1 000	0.685 5	232.42
100	100	200	500	0.679 8	251.08
100	100	200	1 000	0.680 1	232.24
100	200	100	500	0.683 5	168.96
100	200	100	1 000	0.679 8	224.11
100	200	200	500	0.678 3	228.58
100	200	200	1 000	0.663 3	348.89
200	100	100	500	0.670 6	286.63
200	100	100	1 000	0.678 3	227.77

续表 3

第一层隐藏层	第二层隐藏层	第三层隐藏层	迭代次数	准确率	训练时间
200	100	200	500	0.662 5	310.97
200	100	200	1 000	0.684 5	264.18
200	200	100	500	0.676 8	372.08
200	200	100	1 000	0.671 8	467.81
200	200	200	500	0.667 6	403.16
200	200	200	1 000	0.680 2	280.19

按照相关性绝对值从上往下,依次剔除特征发现(见表 4),当同时剔除 id、gender、alco、height、smoke 时,准确率最高,为 0.711 3。

表 4 剔除相关性低的特征模型准确率

相关性	准确率
id(0.003 9)	0.694 2
gender(0.003 9)	0.689 2
alco(0.009 7)	0.6876
height(-0.014)	0.695 7
smoke(-0.019)	0.711 3
active(-0.038)	0.706 7
ap_hi(0.056)	0.690 1
ap_lo(0.062)	0.638 1

最后,使用验证集对分类器进行平均准确率验证,可得验证集 14 663 条数据,准确率为 0.733 1。

4 结束语

在诊断心脑血管疾病的准确率和运行时间见表 5。

表 5 算法准确率总结

算法	准确率	运行时间/s
SVM	0.727 5	56.01
XgBoost	0.751 0	10.657
BP 神经网络	0.711 3	14.23

调节惩罚函数 C,是 SVM 的主要控制因素;XgBoost 的参数调整似乎对结果的影响比较小;剔除合适数量的特征,对 BP 神经网络的准确率有一定的提高;剔除掉一些和标签相关性低的特征,可能会降低准确率。在此实验中,对于 SVM、XgBoost、BP 神经网络三个算法,XgBoost 表现最优。

参考文献:

- [1] 白 宁. 基于主动学习的支持向量机算法[J]. 现代电子技术, 2013, 36(24): 22-28.
- [2] 王 琳. 支持向量机及相关理论研究[D]. 大连: 辽宁师范大学, 2010.
- [3] 何 清, 李 宁, 罗文娟, 等. 大数据下的机器学习算法综述[J]. 模式识别与人工智能, 2014, 27(4): 327-336.
- [4] 朱雄峰, 罗文彩, 魏月兴, 等. 基于结构风险最小化的径向基插值[J]. 弹箭与制导学报, 2011, 31(5): 169-173.
- [5] 张一凡, 冯爱民, 张正林. 支持向量回归增量学习[J]. 计算机科学, 2014, 41(6): 166-170.
- [6] 王 兵. AdaBoost 分类算法的数学分析[J]. 软件, 2014(3): 96-97.
- [7] 朱龙俊, 范君艳. 基于 BP 神经网络的压力传感器误差补偿算法研究[J]. 中国仪器仪表, 2012(9): 32-36.
- [8] 王雪红, 刘晓青, 陶海龙, 等. 优化 BP 神经网络的位移预测模型[J]. 水利水运工程学报, 2014(2): 38-42.
- [9] 陈 欣. 基于 BP 神经网络的学生数学能力评价[J]. 大连教育学院学报, 2013, 29(1): 34-35.