第114天： 三木板模型算法项目实战

原创 戴景波 Python技术 1月14日

机器学习

本篇同样是机器学习，虽然没有用到python中已有的算法和函数，但借鉴了机器学习中的思路。

这篇机器学习建模的思路比较新颖，模型评估也比较独特。旨在引导广大读者借鉴，举一反三。

只是通过足球领域举例，可扩展到其他任何领域，达到抛砖引玉的目的。

机器学习建模

建模思路：选取三家菠菜公司的主队胜赔率（每个公司都会给出初始胜、平、负三个赔率）作为组合对象与历史数据的三家赔率组合比较。

统计出历史比赛结果标签y/n的数量，y代表主队获胜，n代表主队不胜（即主队平或负）。

并输出符合条件“y>nX3或n>yX3”的结果（3为参数，目前尚未调整过此参数）。如果暂未理解思路，也不用担心，马上将结合实例讲解。

这里有两处三，为什么取值为三，这依据了肘部法则(畸变程度的改善效果下降幅度最大的位置就是肘部)，而且经过实践证明了其他诸如二、四、五等值在预测结果准确率上远不如三。可以参考文末的论文。

举例：以下是爬虫爬取的历史数据，每个值都是公司给出的主队第一个赔率也就是获胜的赔率：(以下均以存储在Excel为例)

```
比赛场次  ysb li b5 sna wl ms ao pin y为胜 联赛类型
20191202周日040 1.76 1.75 1.75 1.7 1.78 1.75 1.67 1.82 n  挪超
20191202周日039 1.81 1.7 1.83 1.75 1.8 1.8 1.75 1.75 n  挪超
20191202周日038 1.53 1.57 1.5 1.5 1.52 1.53 0 1.59 y  挪超
20191202周日037 2.03 0 2.1 2.05 2.05 2.03 1.97 2.27 y  挪超
20191202周日036 1.79 1.87 1.75 1.7 1.78 1.79 1.7 1.99 y  挪超
20191202周日035 2.7 2.4 2.88 2.6 2.55 2.58 2.55 2.63 n  挪超
20191202周日031 2.23 2.25 2.25 2.3 2.2 2.25 2.12 2.29 n  德甲
20191202周日029 1.72 1.91 1.73 1.85 1.88 1.85 1.72 1.89 y  英超
20191202周日028 1.53 1.5 1.5 1.5 1.5 1.49 1.48 1.51 y  俄超
20191201周日024 2.25 2.25 2.15 2.3 2.25 2.24 0 2.3 n  西甲
20191201周日023 1.66 1.7 1.67 1.7 1.67 1.69 1.53 1.71 y  德甲
```

下边是未来要预测的一场比赛：

```
20190920周四    2.14    2.15    2.1    2.1    2.2    2.1    2.08    2.12            欧联杯
```

三木板模型算法

选取未来这场比赛中8个特征中的任何三个特征组合（如2.14,2.15,2.1）与历史数据所有行中对应特征（易胜博、立博、bet365）组合进行比对。

发现三个特征值完全相同就统计res列比赛结果y或n的数量。循环其他任意三个特征组合，如(ysb,li,sna)与历史上所有(ysb,li,sna)做精确匹配。

记录当前这场欧联杯比赛在这个组合赔率上历史比赛主队获胜的场次c1和主队不胜的场次c2，如果c1>3倍的c2或c2>3倍的c1，则记录，其他同理，一直到取完56（C83）种组合为止。

也就是说如果历史数据中存在组合对象结果y大于3倍的n时，记录。或n大于3倍的y时，也同样记录。最后根据综合结果来预测未来的这场比赛可能出现的比赛结果。

```
实际结果 信心指数 预测结果 2018.1-至今历史数据 2018.1-至今:欧洲数据 2018.5-至今
y 1.34    y       ,lwa_yes_8_2
```

以上边未来预测的那场比赛为例，最后三列是计算预测结果，其中最后一列得到的结果是lwa_yes_8_2，表示在(li,wl,ao)这三个赔率组合与历史数据比较时，

有8场比赛对应的这三个赔率与这场比赛的赔率相同且结果为主队获胜。有2场比赛对应的这三个赔率与这场比赛相同且结果为主队不胜。

那么yes代表主队获胜，在预测结果列输出y，获胜的信心指数为（8-2X3）X权重，权重会在下边评估模型中讲到。

然而在2018.1-至今历史数据和2018.1-至今:欧洲数据两个阶段的历史数据中，没有输出的结果，表明任何三个赔率的组合在历史数据中都没有主队胜和不胜结果超出3倍的情形，所以不记录。

评估模型

评估模型的建立是为了对建立的机器学习模型进行有效评估，对预测正确的部分进行加强学习，对预测错误的部分进行权重调整，从而达到完善模型的目的。

建立评估模型，旨在选择信心场次，信心场次代表预测的多个比赛中哪些的预测结果出现概率较大。

原理：列出所有C83共56种组合对象，每一列代表一个组合。

用正向激励和反向激励统计出哪些组合对比赛结果有较大的影响，作为今后选择信心场次的优先依据。

```
比赛场次       结果 pre-ok  sum ol5 olin olwl olw olao ol10 o5in o5wl ...
Sum(正向激励)       1为预测正确  373 293 451 363 138 457 353 391
<0（反向激励）     87 50 67 65 31 61 55 87
20190224周日030 y 1  11    3 1   1
  y   11    5    1
  y   9    2 2   1
20190224周日027 y 0  -19
  y   -15
  y   -24   -4   -3
```

第一行为赔率组合（56个组合）；

第二行为正向激励数量，即预测结果正确时各个赔率组合的数量，此时场次对应的值为正数，E列373的公式求和为“=SUM(E4:E65471)”，是所有已预测比赛中(ao,li,b5)组合预测正确的数量。

第三行为反向激励数量，即预测结果错误时各个赔率组合的数量，E列50的公式求和为“=SUM(E4:E65471)”，是所有已预测比赛中(ao,li,b5)组合预测错误的数量。

第四行的最后几列的3、1、1，计算公式为历史数据中比赛结果标签y/n中“多”的数量减去3倍“少”的数量C多-C少X3，如3=4-1X1（历史上存在4场比赛olw三个赔率与这场比赛相同且主队获胜）。

此时场次对应的值为负数；如第九行的-4、-3，计算公式为历史数据中比赛结果标签y/n中多的数量减去3倍少的数量取负值。-（C多-C少X3）。

为进一步量化模型，新增了信心指数中的权重：权重=（>0的数量/sum总数）

```
res 信心指数 预测结果 2018.1-至今历史数据 2018.1-至今:欧洲数据 2018.5-至今
y 1.34  y    ,lw10_yes_8_2
y 9.10  y ,iw10_yes_6_0  ,iw10_yes_4_0  ,iw10_yes_4_0
```

```
changci  lw10 liao10 5inwl 5inw 5inao 5wlw 5in10 ...
sum  151 62 220 279 66 468 166
>0  133 41 143 173 47 237 116
<0  65 13 49 74 15 96 57
权重  0.67 0.76  0.74  0.70 0.76 0.71  0.67
```

以第一行为例，这场比赛的信心指数={历史数据中主队获胜数量-（主队不胜的数量X3） X 对应的权重}：（8-2X3）X 0.67 = 1.34。

评估模型能够进一步量化数据从而得出权重，而这个权重是随着历史数据增加而实时调整的。

同时，评估模型中使用了bagging算法思想，以历史数据中不同阶段作为不同的训练集，例如2018.1-至今历史数据、2018.5-至今等等，分别以不同训练集进行预测。

然后将几个预测结果综合，最终再通过机器学习中的集成学习思想，取最多的结果作为最终预测结果。

所以模型评估对于机器学习非常重要，主要起两个作用：第一、量化权重；第二、反向传播思想改善模型。本篇就很好的完成了这两个作用。

反向传播算法

接下来跟大家详细介绍一下反向传播算法。至于为什么会提出反向传播算法，我直接应用梯度下降（Gradient Descent）不行吗？想必大家肯定有过这样的疑问。

答案肯定是不行的，纵然梯度下降神通广大，但却不是万能的。梯度下降可以应对带有明确求导函数的情况，或者说可以应对那些可以求出误差的情况。

比如逻辑回归（Logistic Regression），我们可以把它看做没有隐层的网络；但对于多隐层的神经网络，输出层可以直接求出误差来更新参数。

但其中隐层的误差是不存在的，因此不能对它直接应用梯度下降，而是先将误差反向传播至隐层，然后再应用梯度下降。

其中将误差从末层往前传递的过程需要链式法则（Chain Rule）的帮助，因此反向传播算法可以说是梯度下降在链式法则中的应用。

在本文中对反向传播的应用则是权重的反向修正，将历史上三个赔率组合对象出现的次数作为total_count，出现yes（主队获胜）数量作为y_count（>0的数量），进而计算出这个赔率组合对象的主队获胜权重y_count/total_count。

随着历史数据的不断增加，权重会一直改变，而且朝着最近时期的趋势方向变化，也正符合业务逻辑。

然后再根据当前比赛得出的结果iw10_yes_4_0计算信心指数，即{历史数据中主队获胜数量-（主队不胜的数量X3） X 对应的权重}。

Bagging算法

Bagging算法的基本思想为给定一个弱学习算法（单个弱学习算法准确率不高）,和一个训练集，将该学习算法使用多次,得出预测函数序列,进行投票，使得最后结果准确率得到提高。

Bagging是并行式集成学习方法的典型代表，它直接基于自助采样法。给定包含m个样本的数据集，我们先随机取出一个样本放入采样中，再把该样本放回初始数据集，使得下次采样时该样本仍有可能被选中。

这样，经过m次随机采样操作，我们得到含m个样本的采样集，初始训练集中有的样本在采样集里多次出现，有的则从未出现。

照这样，我们可采样出T个含m个训练样本的采样集，然后基于每个采样集训练出一个基学习器，再将这些基学习器进行结合。

这就是Bagging的基本流程。在对预测输出进行结合时，Bagging通常对分类任务使用简单投票法，对回归任务使用简单平均法。

若分类预测时出现两个收到同样票数的情形，则最简单的做法是随机选择一个，也可进一步考察学习器投票的置信度来确定最终胜者。

在本文中，借助了Bagging算法的思路，但是采用的是不放回抽样，形成不同阶段的训练数据集，根据每个阶段的投票结果综合计算最后的计算预测结果，

总结

本篇提出了新颖的机器学习预测中的建模思想——三木板模型（已在国家期刊发表论文并被万方数据库收录， 三木板模型算法论文地址：http://wanfangdata.com.cn 搜索“基于机器学习的预测算法模型及其在环评领域的应用”）， 并提出了评估模型的思路，对于足球领域之外的其他领域也非常有借鉴意义，同时对反向传播算法和Bagging算法的原理进行了阐述。

代码地址

> 示例代码：https://github.com/JustDoPython/python-100-day/tree/master/day-100

系列文章

第 113 天：Python XGBoost 算法项目实战
第 112 天：机器学习算法之蒙特卡洛
第 111 天：Python 垃圾回收机制
从 0 学习 Python 0 - 110 大合集总结

PS： 公号内回复 ：Python，即可进入Python 新手学习交流群，一起**100天计划！**

-END-

Python 技术

关于 Python 都在这里