

Welcome

MSRA Researcher新手，现阶段从事多模态大模型预训练方向。对NLP感兴趣或对博客内容有任何疑问及意见建议的，欢迎评论或添加我微信。此外如果有需要内推的同学，也欢迎来骚扰我。联系方式详见concat页面。

前往GitHub (<https://github.com/Dodo>)

统计学习方法 | 逻辑斯蒂原理剖析及实现



12月 3, 2018 (<http://www.pkudodo.com/2018/12/>)

统计学习方法|逻辑斯蒂原理剖析及实现

作者 Dodo (<http://www.pkudodo.com/author/root/>) 在 (<http://www.pkudodo.com/2018/12/03/1-6/>)
机器学习 (<http://www.pkudodo.com/category/%e6%9c%ba%e5%99%a8%e5%ad%a6%e4%b9%a0/>) 标签
实现 (<http://www.pkudodo.com/tag/%e5%ae%9e%e7%8e%b0/>),
统计学习方法
(<http://www.pkudodo.com/tag/%e7%bb%9f%e8%ae%a1%e5%ad%a6%e4%b9%a0%e6%96%b9%e6%b3%95/>),
逻辑斯蒂回归
(<http://www.pkudodo.com/tag/%e9%80%bb%e8%be%91%e6%96%af%e8%92%82%e5%9b%9e%e5%bd%92/>)

阅读数: 14,147

前言

《统计学习方法》一书在前几天正式看完，由于这本书在一定程度上对于初学者是有一些难度的，趁着热乎劲把自己走过的弯路都写出来，后人也能走得更顺畅一点。

以下是我的github地址，其中有《统计学习方法》书中所有涉及到的算法的实现，也是在写博客的同时编写的。在编写宗旨上是希望所有人看完书以后，按照程序的思路再配合书中的公式，能知道怎么讲所学的都应用上。（有点自恋地讲）程序中的注释可能比大部分的博客内容都多。希望大家能够多多捧场，如果可以的话，为我的github打颗星，也能让更多的人看到。

github:

GitHub|手写实现李航《统计学习方法》书中全部算法 (https://github.com/Dod-o/Statistical-Learning-Method_Code)

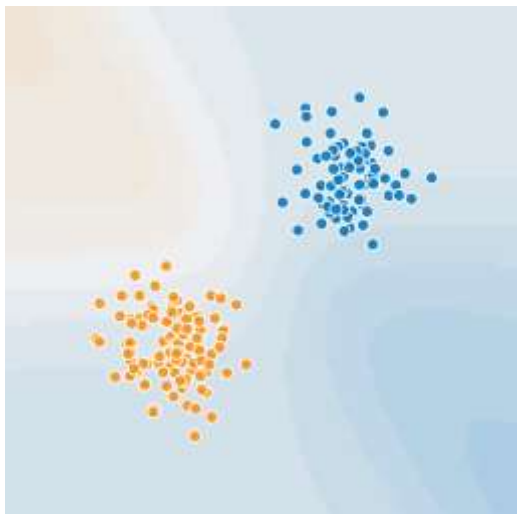
相关博文:

1. 统计学习方法|感知机原理剖析及实现 (<http://www.pkudodo.com/2018/11/18/1-4/>)
2. 统计学习方法|K近邻原理剖析及实现 (<http://www.pkudodo.com/2018/11/19/1-2/>)
3. 统计学习方法|朴素贝叶斯原理剖析及实现 (<http://www.pkudodo.com/2018/11/21/1-3/>)
4. 统计学习方法|决策树原理剖析及实现 (<http://www.pkudodo.com/2018/11/30/1-5/>)
5. 统计学习方法|逻辑斯蒂原理剖析及实现 (<http://www.pkudodo.com/2018/12/03/1-6/>)



正文

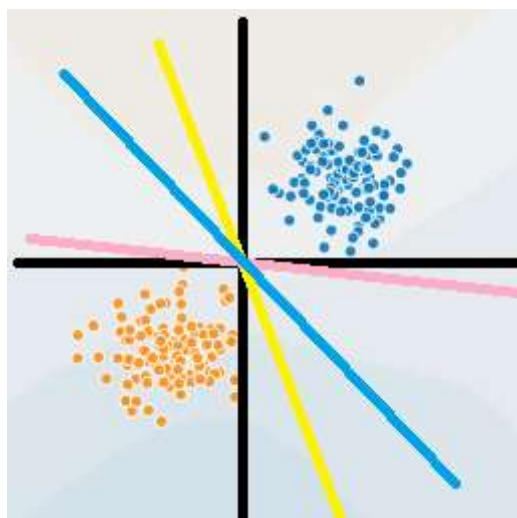
逻辑斯蒂回归的直观理解



(http://www.pkudodo.com/wp-content/uploads/2018/11/K近邻_两堆点.png)

我们还记不记得感知机一章中给出的这张图？还记得感知机是怎么工作的吗？

感知机生成一个超平面（在二维中表现为一条直线），这个超平面可以将图中的两类点区分开。像下面这张图，图中的所有直线都可以作为感知机的超平面（因为它们都能把训练集中的点完美地分开）。



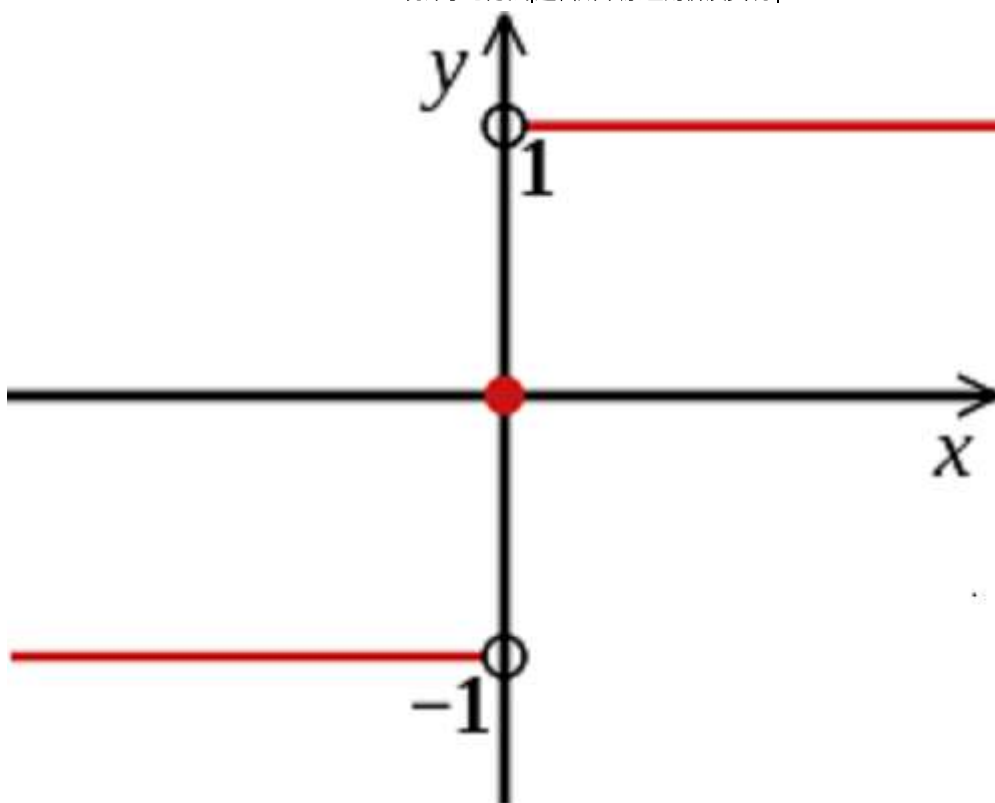
(http://www.pkudodo.com/wp-content/uploads/2018/11/感知机_划分超平面.png)

但是感知机存在一个很重要的问题，我们只用 $\text{sign}(w \cdot x + b)$ 输出的+1和-1来判断点的类别是不是太简单了？是不是有点生硬的感觉。

这么简单的判别方式真的会很有效吗？

当然了，虽然我们已经程序测试过正确率很高，但总是让人有点担心是否在很多情况下都能很好地工作。事实上我们从小到大会听到一些升学考试差一分两分的例子，那么差一分和高一分的学生真的就是天壤之别吗？感知器也是如此：在超平面左侧距离0.001的点和右边0.001距离的点输出就是+1和-1这天壤之别的差距真的合适吗？





(<http://www.pkudodo.com/wp-content/uploads/2018/12/sign图像.png>)

此外我们也知道机器学习中通常会对目标函数进行微分进而梯度下降，但我们看上面这张图。很明显 $x=0$ 是跳跃间断点，因此 sign 是一条不光滑的函数，没有办法进行微分。emmm.....咦？那我记得感知器用了梯度下降，它是怎么去梯度下降的？

我们回忆一下感知器的梯度下降方式，确实用了梯度下降，但是发现没有，它把 sign 的壳子去掉了，对 sign 内部进行梯度下降，相对于其他能直接微分的算法来说，感知器的这种方式确实有点不太好。

算法 2.1（感知机学习算法的原始形式）

输入：训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ ，其中 $x_i \in \mathcal{X} = \mathbf{R}^n$ ， $y_i \in \mathcal{Y} = \{-1, +1\}$ ， $i = 1, 2, \dots, N$ ；学习率 η ($0 < \eta \leq 1$)；

输出： w, b ；感知机模型 $f(x) = \text{sign}(w \cdot x + b)$ 。

(1) 选取初值 w_0, b_0

(2) 在训练集中选取数据 (x_i, y_i)

(3) 如果 $y_i(w \cdot x_i + b) \leq 0$

$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b + \eta y_i$$

(4) 转至 (2)，直至训练集中没有误分类点。 ■

(<http://www.pkudodo.com/wp-content/uploads/2018/12/感知器算法.png>)

因此既然我们要对感知器进行优化，那么上文的这两个主要的缺陷咱们得想想法子看看能不能弥补。我们首先试试解决第一个问题：

1. 怎么解决极小距离差别带来的+1和-1的天壤之别？

在逻辑斯蒂回归中大致思想与感知器相同，但在输出+1与-1之间存在一些差别。在朴素贝叶斯中我们提到过 $P(Y|X)$ ，它表示在给定 X 条件下， Y 发生的概率。逻辑斯蒂也使用了同样的概念，它使用 $p(Y=-1|X)$ 和 $p(Y=1|X)$ 来表示该样本分别为-1或1的概率（实际上逻辑斯蒂并非强制要求标签必须为1或-1，可以用任意标签来表示）。这样当再出现样本 X_1 距离为-0.001时，可能 $P(Y=1|X_1)=0.49$ ， $P(Y=0|X_1)=0.51$ ，那么我们觉得 X_1 为0的概率更大一点，但我们同时也清楚程序可能并不太确定 X_1 一定为0。

使用概率作为输出结果使得样本在距离很小的差别下不再强制地输出+1和-1这两种天壤之别的结果，而是通过概率的方式告诉你结果可能是多少，同时也告诉你预测的不确信程度。这样子看起来让人比较安心一点不是吗？

2.怎么让最终的预测式子可微呢？

虽然无法微分并不会阻碍感知器的正常工作（事实上只是避开了sign），但对于很多场合都需要微分的机器学习来说，能找到一个可以微分的最终式子是很重要的。为了解决第一个问题我们提出了一种概率输出模型，那么感知器的sign也需要被随之替换为一种能输出概率大小的函数。具体函数在下文中详细讲解，其中值得高兴的是，我们找到的概率输出式子是平滑的，可微的，所以第二个问题也就迎刃而解了。

逻辑斯蒂回归的数学角度（配合《统计学习方法》食用更佳）

与感知器一样，我们先不管内部怎么工作，最好能够得到一个函数 $f(x)$ ，我们将样本 x 放进去以后，它告诉我们属于1的概率是多少。比如说 $f(y=1|x)=0.2$ ， $f(y=0|x)=0.8$ ，我们就知道该样本的标签大概率是0。总结一下也就是分别比较两种类别的概率大小，概率大的那一方则作为预测类别进行输出。 f 函数的定义如下图所示。

定义 6.2（逻辑斯蒂回归模型） 二项逻辑斯蒂回归模型是如下的条件概率分布：

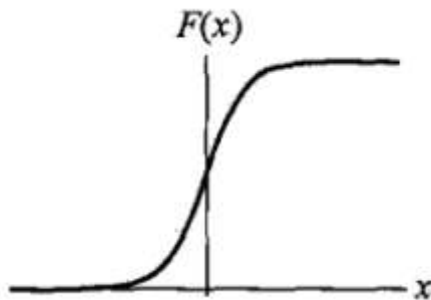
$$P(Y=1|x) = \frac{\exp(w \cdot x + b)}{1 + \exp(w \cdot x + b)} \quad (6.3)$$

$$P(Y=0|x) = \frac{1}{1 + \exp(w \cdot x + b)} \quad (6.4)$$

这里， $x \in \mathbf{R}^n$ 是输入， $Y \in \{0,1\}$ 是输出， $w \in \mathbf{R}^n$ 和 $b \in \mathbf{R}$ 是参数， w 称为权值向量， b 称为偏置， $w \cdot x$ 为 w 和 x 的内积。

(http://www.pkudodo.com/wp-content/uploads/2018/12/逻辑斯蒂回归_定义6.2.png)

下图就是 $f(wx+b)$ 的图像，我们假设一下点 X 在超平面右边，那么距离应当为正，如果距离无穷大时， $\exp(w \cdot x + b)$ 无穷大， $P(Y=1|x)=1$ ，也就是概率为1，表极其确信。如果 $w \cdot x + b$ 是一个很接近0的数， $\exp(w \cdot x + b)$ 接近1， $P(Y=1|x)=0.5$ ，表示两边都有可能，不太确定。我们对于每一个样本点分别计算属于两类的概率，概率较大的一方作为预测的输出。



(<http://www.pkudodo.com/wp-content/uploads/2018/12/sigmoid图像.png>)

既然模型有了，那么就需要计算参数以此来得到划分的超平面。在计算参数上一般使用最大似然函数，关于最大似然函数的介绍可以在网上进行相关浏览。由于我们使用的sigmoid函数很光滑，是可微函数，因此可以直接对似然函数求导，进行梯度下降找到参数的最优点。

逻辑斯蒂回归模型学习时，对于给定的训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ ，其中， $x_i \in \mathbf{R}^n$ ， $y_i \in \{0, 1\}$ ，可以应用极大似然估计法估计模型参数，从而得到逻辑斯蒂回归模型。

设： $P(Y=1|x) = \pi(x)$ ， $P(Y=0|x) = 1 - \pi(x)$

似然函数为

$$\prod_{i=1}^N [\pi(x_i)]^{y_i} [1 - \pi(x_i)]^{1-y_i}$$

对数似然函数为

$$\begin{aligned} L(w) &= \sum_{i=1}^N [y_i \log \pi(x_i) + (1 - y_i) \log(1 - \pi(x_i))] \\ &= \sum_{i=1}^N \left[y_i \log \frac{\pi(x_i)}{1 - \pi(x_i)} + \log(1 - \pi(x_i)) \right] \\ &= \sum_{i=1}^N [y_i (w \cdot x_i) - \log(1 + \exp(w \cdot x_i))] \end{aligned}$$

对 $L(w)$ 求极大值，得到 w 的估计值。

(<http://www.pkudodo.com/wp-content/uploads/2018/12/逻辑斯蒂回归.png>)关于似然函数关于 w 的求导式可以看下图：

$$\begin{aligned} L(w) &= \sum_{i=1}^N [y_i (w \cdot x_i) - \log(1 + \exp(w \cdot x_i))] \\ \frac{\partial L(w)}{\partial w_i} &= y_i x_i - \frac{1}{1 + \exp(w \cdot x_i)} \cdot \exp(w \cdot x_i) \cdot x_i \\ &= y_i x_i - \frac{x_i \exp(w \cdot x_i)}{1 + \exp(w \cdot x_i)} \end{aligned}$$

(http://www.pkudodo.com/wp-content/uploads/2018/12/逻辑斯蒂_梯度下降.png)

^

结语：逻辑斯蒂在思想上与感知器极为相近，很多细节问题在感知器中已有说明因此在本文中并未详细展开。如果对部分地方有疑问可查看感知器一章节，此外也可以评论或邮箱的形式联系我。

贴代码：（建议去本文最上方的github链接下载，有书中所有算法的实现以及详细注释）

```

1. # coding=utf-8
2. # Author:Dodo
3. # Date:2018-11-27
4. # Email:lvtingchao@pku.edu.cn
5. # Blog:www.pkudodo.com
6.
7. '''
8. 数据集: Mnist
9. 训练集数量: 60000
10. 测试集数量: 10000
11. -----
12. 运行结果:
13.     正确率: 98.91%
14.     运行时长: 59s
15. '''
16.
17. import time
18. import numpy as np
19.
20.
21. def loadData(fileName):
22.     '''
23.     加载Mnist数据集
24.     :param fileName:要加载的数据集路径
25.     :return: list形式的数据集及标记
26.     '''
27.     # 存放数据及标记的list
28.     dataList = []; labelList = []
29.     # 打开文件
30.     fr = open(fileName, 'r')
31.     # 将文件按行读取
32.     for line in fr.readlines():
33.         # 对每一行数据按切割福','进行切割，返回字段列表
34.         curLine = line.strip().split(',')
35.
36.         # Mnsit有0-9是个标记，由于是二分类任务，所以将标记0的作为
1, 其余为0
37.         # 验证过<5为1 >5为0时正确率在90%左右，猜测是因为数多了以
后，可能不同数的特征较乱，不能有效地计算出一个合理的超平面
38.         # 查看了一下之前感知机的结果，以5为分界时正确率81，重新修改
为0和其余数时正确率98.91%

```



```

39.         # 看来如果样本标签比较杂的话，对于是否能有效地划分超平面确实存在很大影响
40.         if int(curLine[0]) == 0:
41.             labellist.append(1)
42.         else:
43.             labellist.append(0)
44.         #存放标记
45.         #[int(num) for num in curLine[1:]] -> 遍历每一行中除了以第一哥元素（标记）外将所有元素转换成int类型
46.         #[int(num)/255 for num in curLine[1:]] -> 将所有数据除255归一化(非必须步骤，可以不归一化)
47.         dataList.append([int(num)/255 for num in curLine[1:]])
48.         # dataList.append([int(num) for num in curLine[1:]])
49.
50.     #返回data和label
51.     return dataList, labellist
52.
53. def predict(w, x):
54.     '''
55.     预测标签
56.     :param w: 训练过程中学到的w
57.     :param x: 要预测的样本
58.     :return: 预测结果
59.     '''
60.     #dot为两个向量的点积操作，计算得到w * x
61.     wx = np.dot(w, x)
62.     #计算标签为1的概率
63.     #该公式参考“6.1.2 二项逻辑斯蒂回归模型”中的式6.5
64.     P1 = np.exp(wx) / (1 + np.exp(wx))
65.     #如果为1的概率大于0.5，返回1
66.     if P1 >= 0.5:
67.         return 1
68.     #否则返回0
69.     return 0
70.
71. def logisticRegression(trainDataList, trainLabellist, iter = 200):
72.     '''
73.     逻辑斯蒂回归训练过程
74.     :param trainDataList: 训练集
75.     :param trainLabellist: 标签集
76.     :param iter: 迭代次数
77.     :return: 习得的w
78.     '''
79.     #按照书本“6.1.2 二项逻辑斯蒂回归模型”中式6.5的规则，将w与b合在一起，
80.     #此时x也需要添加一维，数值为1
81.     #循环遍历每一个样本，并在其最后添加一个1
82.     for i in range(len(trainDataList)):

```

^


```

83.         trainDataList[i].append(1)
84.
85.         #将数据集由列表转换为数组形式，主要是后期涉及到向量的运算，统一
转换成数组形式比较方便
86.         trainDataList = np.array(trainDataList)
87.         #初始化w，维数为样本x维数+1，+1的那一位是b，初始为0
88.         w = np.zeros(trainDataList.shape[1])
89.
90.         #设置步长
91.         h = 0.001
92.
93.         #迭代iter次进行随机梯度下降
94.         for i in range(iter):
95.             #每次迭代冲遍历一次所有样本，进行随机梯度下降
96.             for j in range(trainDataList.shape[0]):
97.                 #随机梯度上升部分
98.                 #在“6.1.3 模型参数估计”一章中给出了似然函数，我们需要极
大化似然函数
99.                 #但是似然函数由于有求和项，并不能直接对w求导得出最优w，
所以针对似然函数求和
100.                #部分中每一项进行单独地求导w，得到针对该样本的梯度，并
进行梯度上升（因为是
101.                #要求似然函数的极大值，所以是梯度上升，如果是极小值就
梯度下降。梯度上升是
102.                #加号，下降是减号）
103.                #求和式中每一项单独对w求导结果为：xi * yi - (exp(w *
xi) * xi) / (1 + exp(w * xi))
104.                #如果对于该求导式有疑问可查看我的博客 www.pkudodo.com
105.
106.                #计算w * xi，因为后式中要计算两次该值，为了节约时间这
里提前算出
107.                #其实也可直接算出exp(wx)，为了读者能看得方便一点就这么
写了，包括yi和xi都提前列出了
108.                wx = np.dot(w, trainDataList[j])
109.                yi = trainLabelList[j]
110.                xi = trainDataList[j]
111.                #梯度上升
112.                w += h * (xi * yi - (np.exp(wx) * xi) / (1 +
np.exp(wx)))
113.
114.            #返回学到的w
115.            return w
116.
117. def test(testDataList, testLabelList, w):
118.     '''
119.     验证
120.     :param testDataList:测试集
121.     :param testLabelList: 测试集标签
122.     :param w: 训练过程中学到的w

```



```
123.         :return: 正确率
124.         '''
125.
126.     #与训练过程一致，先将所有的样本添加一维，值为1，理由请查看训练
函数
127.     for i in range(len(testDataList)):
128.         testDataList[i].append(1)
129.
130.     #错误值计数
131.     errorCnt = 0
132.     #对于测试集中每一个测试样本进行验证
133.     for i in range(len(testDataList)):
134.         #如果标记与预测不一致，错误值加1
135.         if testLabelList[i] != predict(w, testDataList[i]):
136.             errorCnt += 1
137.     #返回准确率
138.     return 1 - errorCnt / len(testDataList)
139.
140.
141.
142. if __name__ == '__main__':
143.     start = time.time()
144.
145.     # 获取训练集及标签
146.     print('start read transSet')
147.     trainData, trainLabel = loadData('../Mnist/mnist_train.csv')
148.
149.     # 获取测试集及标签
150.     print('start read testSet')
151.     testData, testLabel = loadData('../Mnist/mnist_test.csv')
152.
153.     # 开始训练，学习w
154.     print('start to train')
155.     w = logisticRegression(trainData, trainLabel)
156.
157.     #验证正确率
158.     print('start to test')
159.     accuracy = test(testData, testLabel, w)
160.
161.     # 打印准确率
162.     print('the accuracy is:', accuracy)
163.     # 打印时间
164.     print('time span:', time.time() - start)
```





Dodo

💬 13条评论



匿名

发布于6:32 上午 - 1月 28, 2023

To the pkudodo.com owner, Keep up the good work, admin!

()

回复



匿名

发布于5:13 下午 - 1月 9, 2023

Just would like to say your posting is as incredible. The clearness as part of your write-up is actually nice and i can assume you're a professional on this matter. Effectively with all your permission allow me to to grab your RSS feed to help keep current with forthcoming submit. Thanks 1,000,000 and make sure you keep on the pleasurable work. sasilu.se/map8.php telia och viasat

()

回复



匿名

发布于5:09 下午 - 1月 9, 2023

You will be so attention-grabbing! I don't suppose I've really browse by one thing like that ahead of. So good to find out A different man or woman with a few unique views on this subject. Seriously.. thanks for starting off this up. This page is another thing that is needed on the web, anyone with some originality!

()

回复

匿名

发布于11:34 下午 - 9月 6, 2022



bunadisisj nsjjsjsjsmizjzjjz zumzsert

回复

()



匿名

发布于10:11 上午 - 11月 23, 2021

91lgxh

回复

()



匿名

发布于9:32 下午 - 10月 11, 2020

您好，怎么能联系上呢？有问题请教下

回复

()



匿名

发布于4:00 下午 - 3月 20, 2020

请问下逻辑斯蒂回归的解析解怎么求？

回复

()



匿名

发布于4:25 下午 - 2月 16, 2020

手动对 w 求导那个应该是手误写错了， w 的下标用其他字母表示，要不然就乱了

回复

()



匿名

发布于5:04 下午 - 7月 5, 2020

你好我没听明白 可以详细讲讲吗

()



回复



匿名

发布于4:01 下午 - 11月 12, 2019

()

想问一下为什么把下边的代码

```
if(curline[0]=='0'):# 二分类问题 转换为区分0和非0数字
dataY.append(1) # 标签
```

else:

```
dataY.append(0)
```

换成:

```
if(curline[0]=='0'):# 二分类问题 转换为区分0和非0数字
dataY.append(0) # 标签
```

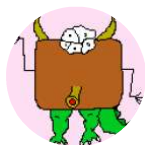
else:

```
dataY.append(1)
```

然后正确率就直接为原来的错误率了

感觉标签改变应该不影响预测结果吧?

回复



匿名

发布于7:45 下午 - 11月 12, 2019

()

突然发现上述情况的发生是在dataX.append([int(num)for num in curline[1:]]) 未进行归一化的前提下

进行归一化后, 上述情况就会消失...

不太清楚原因为何。

回复



匿名

发布于3:11 下午 - 9月 3, 2019

()

你好 这个逻辑回归没有实现正则化吗?

回复



匿名

发布于2:59 下午 - 2月 7, 2019

楼主, 请问什么时候添加GBDT和XGboost的原理和源码分析, 谢谢分享



()

回复

发表评论

内容(链接请去除http协议头，否则会被误判为垃圾评论)

设置显示昵称(选填)

发送信息

近期文章

- ✓ 论文 | 记忆网络之Memory Networks (<http://www.pkudodo.com/2019/06/14/1-13/>)
- ✓ 梳理 | 对话系统中的DST (<http://www.pkudodo.com/2019/06/09/1-12/>)
- ✓ 论文阅读 | How Does Batch Normalization Help Optimization (<http://www.pkudodo.com/2019/05/23/1-11/>)
- ✓ 学习规划 | 机器学习和NLP入门规划 (<http://www.pkudodo.com/2019/03/20/1-10/>)
- ✓ 面试体会 | 微软、头条、滴滴、爱奇艺NLP面试感想 (<http://www.pkudodo.com/2019/03/10/1-9/>)



文章归档

- ✓ 2019年6月 (<http://www.pkudodo.com/2019/06/>)
- ✓ 2019年5月 (<http://www.pkudodo.com/2019/05/>)
- ✓ 2019年3月 (<http://www.pkudodo.com/2019/03/>)
- ✓ 2018年12月 (<http://www.pkudodo.com/2018/12/>)
- ✓ 2018年11月 (<http://www.pkudodo.com/2018/11/>)
- ✓ 2018年10月 (<http://www.pkudodo.com/2018/10/>)
- ✓ 2016年9月 (<http://www.pkudodo.com/2016/09/>)

标签

DST (<http://www.pkudodo.com/tag/dst/>)

K近邻 (<http://www.pkudodo.com/tag/k%E8%BF%91%E9%82%BB/>)

LSI (<http://www.pkudodo.com/tag/lsi/>)

memory network (<http://www.pkudodo.com/tag/memory-network/>)

MQTT (<http://www.pkudodo.com/tag/mqtt/>)

NLP (<http://www.pkudodo.com/tag/nlp/>)

s3c2440 (<http://www.pkudodo.com/tag/s3c2440/>)

SLU (<http://www.pkudodo.com/tag/slu/>)

SVM (<http://www.pkudodo.com/tag/svm/>)

VSM (<http://www.pkudodo.com/tag/vsm/>)

体会 (<http://www.pkudodo.com/tag/%E4%BD%93%E4%BC%9A/>)

决策树 (<http://www.pkudodo.com/tag/%E5%86%B3%E7%AD%96%E6%A0%91/>)

分类器 (<http://www.pkudodo.com/tag/%E5%88%86%E7%B1%BB%E5%99%A8/>)

实现 (<http://www.pkudodo.com/tag/%E5%AE%9E%E7%8E%B0/>)



对话系统 (<http://www.pkudodo.com/tag/%e5%af%b9%e8%af%9d%e7%b3%bb%e7%bb%9f/>)

感想 (<http://www.pkudodo.com/tag/%e6%84%9f%e6%83%b3/>)

感知机 (<http://www.pkudodo.com/tag/%e6%84%9f%e7%9f%a5%e6%9c%ba/>)

支持向量机 (<http://www.pkudodo.com/tag/%e6%94%af%e6%8c%81%e5%90%91%e9%87%8f%e6%9c%ba/>)

文章相似度 (<http://www.pkudodo.com/tag/%e6%96%87%e7%ab%a0%e7%9b%b8%e4%bc%bc%e5%ba%a6/>)

最大熵 (<http://www.pkudodo.com/tag/%e6%9c%80%e5%a4%a7%e7%86%b5/>)

朴素贝叶斯 (<http://www.pkudodo.com/tag/%e6%9c%b4%e7%b4%a0%e8%b4%9d%e5%8f%b6%e6%96%af/>)

机器学习 (<http://www.pkudodo.com/tag/%e6%9c%ba%e5%99%a8%e5%ad%a6%e4%b9%a0/>)

爬虫 (<http://www.pkudodo.com/tag/%e7%88%ac%e8%99%ab/>)

统计学习方法 (<http://www.pkudodo.com/tag/%e7%bb%9f%e8%ae%a1%e5%ad%a6%e4%b9%a0%e6%96%b9%e>

统计方法学习 (<http://www.pkudodo.com/tag/%e7%bb%9f%e8%ae%a1%e6%96%b9%e6%b3%95%e5%ad%a6%e>

综述 (<http://www.pkudodo.com/tag/%e7%bb%bc%e8%bf%b0/>)

网易云歌单 (<http://www.pkudodo.com/tag/%e7%bd%91%e6%98%93%e4%ba%91%e6%ad%8c%e5%8d%95/>)

规划 (<http://www.pkudodo.com/tag/%e8%a7%84%e5%88%92/>)

详解 (<http://www.pkudodo.com/tag/%e8%af%a6%e8%a7%a3/>)

逻辑回归 (<http://www.pkudodo.com/tag/%e9%80%bb%e8%be%91%e5%9b%9e%e5%bd%92/>)

逻辑斯蒂回归 (<http://www.pkudodo.com/tag/%e9%80%bb%e8%be%91%e6%96%af%e8%92%82%e5%9b%9e%e>

面试 (<http://www.pkudodo.com/tag/%e9%9d%a2%e8%af%95/>)

