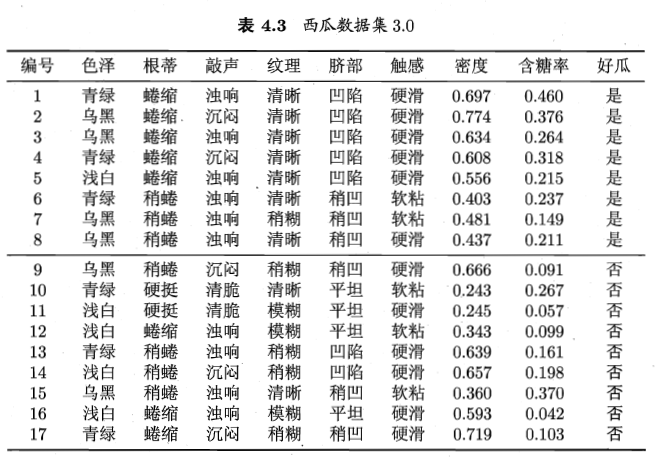
# 朴素贝叶斯分类器

1. 模型概述
2. 概念描述

朴素贝叶斯分类器是**一系列以假设特征之间强（朴素）独立下运用 贝叶斯定理 为基础的简单概率分类器**。该分类器模型会给问题实例分配用特征值表示的类标签，类标签取自有限集合。. 它不是训练这种分类器的单一算法，而是一系列基于相同原理的算法：所有朴素贝叶斯分类器都假定样本每个特征与其他特征都不相关。

1. 公式描述

数据集：



公式：

|  |  |  |
| --- | --- | --- |
| **先验概率(数据为离散值)** |  | |
| **后验概率 (数据为离散值)** |  | |
| **后验概率 (数据为连续值)** | (分别为第c类样本在第i个属性上取值的均值和方差。) | |
| **目标值** |  | |
| **拉普拉斯修正** |  |  |

1. 重要代码详解

求连续值概率：传进来的参数ave和std分别为该样本在该属性上的均值和标准差，则left为，right为，返回值保留了3位小数。

def Continuous\_p(x,ave,std):

    left=1/(np.sqrt(2\*math.pi)\*std)

    right=np.exp((-np.power(x-ave,2))/(2\*np.power(std,2)))

    return round(left\*right,3)

求先验概率和后验概率：设置循环对测试样本中的属性进行计数并且计算概率，最后返回所有概率值。这段代码采用了拉普拉斯修正，源码中同样有未使用拉普拉斯修正的测试数据，原理一样，只是计算概率方式不同。

def Bayes(data,test):

    d={'好瓜':0,'坏瓜':0,'好瓜'+test[0]:0,'坏瓜'+test[0]:0,'好瓜'+test[1]:0,'坏瓜'+test[1]:0,'好瓜'+test[2]:0,'坏瓜'+test[2]:0,

    '好瓜'+test[3]:0,'坏瓜'+test[3]:0,'好瓜'+test[4]:0,'坏瓜'+test[4]:0,'好瓜'+test[5]:0,'坏瓜'+test[5]:0,

    '好瓜密度ave':0,'好瓜密度var':0,'好瓜含糖率ave':0,'好瓜含糖率var':0,'坏瓜密度ave':0,'坏瓜密度var':0,'坏瓜含糖率ave':0,'坏瓜含糖率var':0}

    good\_density=[]

    bad\_density=[]

    good\_sugar=[]

    bad\_sugar=[]

    for item in np.array(data):

        if item[8]=='是':

            d['好瓜']+=1

            good\_density.append(item[6])

            good\_sugar.append(item[7])

            if item[0]==test[0]:

                d['好瓜'+test[0]]+=1

            if item[1]==test[1]:

                d['好瓜'+test[1]]+=1

            if item[2]==test[2]:

                d['好瓜'+test[2]]+=1

            if item[3]==test[3]:

                d['好瓜'+test[3]]+=1

            if item[4]==test[4]:

                d['好瓜'+test[4]]+=1

            if item[5]==test[5]:

                d['好瓜'+test[5]]+=1

        elif item[8]=='否':

            d['坏瓜']+=1

            bad\_density.append(item[6])

            bad\_sugar.append(item[7])

            if item[0]==test[0]:

                d['坏瓜'+test[0]]+=1

            if item[1]==test[1]:

                d['坏瓜'+test[1]]+=1

            if item[2]==test[2]:

                d['坏瓜'+test[2]]+=1

            if item[3]==test[3]:

                d['坏瓜'+test[3]]+=1

            if item[4]==test[4]:

                d['坏瓜'+test[4]]+=1

            if item[5]==test[5]:

                d['坏瓜'+test[5]]+=1

    d['好瓜密度ave']=np.average(good\_density)

    d['好瓜密度var']=np.std(good\_density,ddof=1)

    d['好瓜含糖率ave']=np.average(good\_sugar)

    d['好瓜含糖率var']=np.std(good\_sugar,ddof=1)

    d['坏瓜密度ave']=np.average(bad\_density)

    d['坏瓜密度var']=np.std(bad\_density,ddof=1)

    d['坏瓜含糖率ave']=np.average(bad\_sugar)

    d['坏瓜含糖率var']=np.std(bad\_sugar,ddof=1)

    num=len(np.unique(list(data['好瓜'])))

    num0=len(np.unique(list(data['色泽'])))

    num1=len(np.unique(list(data['根蒂'])))

    num2=len(np.unique(list(data['敲声'])))

    num3=len(np.unique(list(data['纹理'])))

    num4=len(np.unique(list(data['脐部'])))

    num5=len(np.unique(list(data['触感'])))

    p\_1\_p=round((d['好瓜']+1)/(len(data['好瓜'])+num),3)

    p\_0\_p=round((d['坏瓜']+1)/(len(data['好瓜'])+num),3)

    p\_x0\_1=round((d['好瓜'+test[0]]+1)/(d['好瓜']+num0),3)

    p\_x0\_0=round((d['坏瓜'+test[0]]+1)/(d['坏瓜']+num0),3)

    p\_x1\_1=round((d['好瓜'+test[1]]+1)/(d['好瓜']+num1),3)

    p\_x1\_0=round((d['坏瓜'+test[1]]+1)/(d['坏瓜']+num1),3)

    p\_x2\_1=round((d['好瓜'+test[2]]+1)/(d['好瓜']+num2),3)

    p\_x2\_0=round((d['坏瓜'+test[2]]+1)/(d['坏瓜']+num2),3)

    p\_x3\_1=round((d['好瓜'+test[3]]+1)/(d['好瓜']+num3),3)

    p\_x3\_0=round((d['坏瓜'+test[3]]+1)/(d['坏瓜']+num3),3)

    p\_x4\_1=round((d['好瓜'+test[4]]+1)/(d['好瓜']+num4),3)

    p\_x4\_0=round((d['坏瓜'+test[4]]+1)/(d['坏瓜']+num4),3)

    p\_x5\_1=round((d['好瓜'+test[5]]+1)/(d['好瓜']+num5),3)

    p\_x5\_0=round((d['坏瓜'+test[5]]+1)/(d['坏瓜']+num5),3)

    p\_x6\_1=Continuous\_p(test[6],d['好瓜密度ave'],d['好瓜密度var'])

    p\_x6\_0=Continuous\_p(test[6],d['坏瓜密度ave'],d['坏瓜密度var'])

    p\_x7\_1=Continuous\_p(test[7],d['好瓜含糖率ave'],d['好瓜含糖率var'])

    p\_x7\_0=Continuous\_p(test[7],d['坏瓜含糖率ave'],d['坏瓜含糖率var'])

    return p\_0\_p,p\_1\_p,p\_x0\_0,p\_x0\_1,p\_x1\_0,p\_x1\_1,p\_x2\_0,p\_x2\_1,p\_x3\_0,p\_x3\_1,p\_x4\_0,p\_x4\_1,p\_x5\_0,p\_x5\_1,p\_x6\_0,p\_x6\_1,p\_x7\_0,p\_x7\_1

1. 结果分析

当采用测试样例为[青绿，蜷缩，浊响，清晰，凹陷，硬滑，0.697，0.460]时不采用拉普拉斯修正也是可以的，因为不管好瓜坏瓜其中都至少有一种上述属性，不会导致其中一种后验概率为0.最后结果检测该样例为好瓜，且是好瓜的概率为0.05242077566070558 是坏瓜的概率为6.797801125853964e-05，相差较大，所以该测试样例取得了比较好的结果。

如果采用测试样例为[青绿，蜷缩，清脆，清晰，凹陷，硬滑，0.697，0.460]其中好瓜中没有清脆，所以会导致最后相乘为0，则必须使用拉普拉斯修正，最后结果为好瓜，是好瓜的概率为0.0036635930507095235是坏瓜的概率为 4.602151547019676e-05，相差较大，该测试样例结果较好。

四.源码

#判断一个具有特征{色泽=青绿,根蒂=蜷缩,敲声=浊响,纹理=清晰,脐部=凹陷,触感=硬滑,密度=0.697,含糖率=0.460}的测试样例瓜是否为好瓜。

import math

import numpy as np

import pandas as pd

def Continuous\_p(x,ave,std):

    left=1/(np.sqrt(2\*math.pi)\*std)

    right=np.exp((-np.power(x-ave,2))/(2\*np.power(std,2)))

    return round(left\*right,3)

#测试样例

test=['青绿','蜷缩','浊响','清晰','凹陷','硬滑',0.697,0.460]

data=pd.read\_csv("朴素贝叶斯分类器\data.txt",names=['色泽','根蒂','敲声','纹理','脐部','触感','密度','含糖率','好瓜'])

def Bayes(data,test):

    d={'好瓜':0,'坏瓜':0,'好瓜'+test[0]:0,'坏瓜'+test[0]:0,'好瓜'+test[1]:0,'坏瓜'+test[1]:0,'好瓜'+test[2]:0,'坏瓜'+test[2]:0,

    '好瓜'+test[3]:0,'坏瓜'+test[3]:0,'好瓜'+test[4]:0,'坏瓜'+test[4]:0,'好瓜'+test[5]:0,'坏瓜'+test[5]:0,

    '好瓜密度ave':0,'好瓜密度var':0,'好瓜含糖率ave':0,'好瓜含糖率var':0,'坏瓜密度ave':0,'坏瓜密度var':0,'坏瓜含糖率ave':0,'坏瓜含糖率var':0}

    good\_density=[]

    bad\_density=[]

    good\_sugar=[]

    bad\_sugar=[]

    for item in np.array(data):

        if item[8]=='是':

            d['好瓜']+=1

            good\_density.append(item[6])

            good\_sugar.append(item[7])

            if item[0]==test[0]:

                d['好瓜'+test[0]]+=1

            if item[1]==test[1]:

                d['好瓜'+test[1]]+=1

            if item[2]==test[2]:

                d['好瓜'+test[2]]+=1

            if item[3]==test[3]:

                d['好瓜'+test[3]]+=1

            if item[4]==test[4]:

                d['好瓜'+test[4]]+=1

            if item[5]==test[5]:

                d['好瓜'+test[5]]+=1

        elif item[8]=='否':

            d['坏瓜']+=1

            bad\_density.append(item[6])

            bad\_sugar.append(item[7])

            if item[0]==test[0]:

                d['坏瓜'+test[0]]+=1

            if item[1]==test[1]:

                d['坏瓜'+test[1]]+=1

            if item[2]==test[2]:

                d['坏瓜'+test[2]]+=1

            if item[3]==test[3]:

                d['坏瓜'+test[3]]+=1

            if item[4]==test[4]:

                d['坏瓜'+test[4]]+=1

            if item[5]==test[5]:

                d['坏瓜'+test[5]]+=1

    d['好瓜密度ave']=np.average(good\_density)

    d['好瓜密度var']=np.std(good\_density,ddof=1)

    d['好瓜含糖率ave']=np.average(good\_sugar)

    d['好瓜含糖率var']=np.std(good\_sugar,ddof=1)

    d['坏瓜密度ave']=np.average(bad\_density)

    d['坏瓜密度var']=np.std(bad\_density,ddof=1)

    d['坏瓜含糖率ave']=np.average(bad\_sugar)

    d['坏瓜含糖率var']=np.std(bad\_sugar,ddof=1)

    p\_1\_p=round(d['好瓜']/len(data['好瓜']),3)

    p\_0\_p=round(d['坏瓜']/len(data['好瓜']),3)

    p\_x0\_1=round(d['好瓜'+test[0]]/d['好瓜'],3)

    p\_x0\_0=round(d['坏瓜'+test[0]]/d['坏瓜'],3)

    p\_x1\_1=round(d['好瓜'+test[1]]/d['好瓜'],3)

    p\_x1\_0=round(d['坏瓜'+test[1]]/d['坏瓜'],3)

    p\_x2\_1=round(d['好瓜'+test[2]]/d['好瓜'],3)

    p\_x2\_0=round(d['坏瓜'+test[2]]/d['坏瓜'],3)

    p\_x3\_1=round(d['好瓜'+test[3]]/d['好瓜'],3)

    p\_x3\_0=round(d['坏瓜'+test[3]]/d['坏瓜'],3)

    p\_x4\_1=round(d['好瓜'+test[4]]/d['好瓜'],3)

    p\_x4\_0=round(d['坏瓜'+test[4]]/d['坏瓜'],3)

    p\_x5\_1=round(d['好瓜'+test[5]]/d['好瓜'],3)

    p\_x5\_0=round(d['坏瓜'+test[5]]/d['坏瓜'],3)

    p\_x6\_1=Continuous\_p(test[6],d['好瓜密度ave'],d['好瓜密度var'])

    p\_x6\_0=Continuous\_p(test[6],d['坏瓜密度ave'],d['坏瓜密度var'])

    p\_x7\_1=Continuous\_p(test[7],d['好瓜含糖率ave'],d['好瓜含糖率var'])

    p\_x7\_0=Continuous\_p(test[7],d['坏瓜含糖率ave'],d['坏瓜含糖率var'])

    return p\_0\_p,p\_1\_p,p\_x0\_0,p\_x0\_1,p\_x1\_0,p\_x1\_1,p\_x2\_0,p\_x2\_1,p\_x3\_0,p\_x3\_1,p\_x4\_0,p\_x4\_1,p\_x5\_0,p\_x5\_1,p\_x6\_0,p\_x6\_1,p\_x7\_0,p\_x7\_1

p\_0\_p,p\_1\_p,p\_x0\_0,p\_x0\_1,p\_x1\_0,p\_x1\_1,p\_x2\_0,p\_x2\_1,p\_x3\_0,p\_x3\_1,p\_x4\_0,p\_x4\_1,p\_x5\_0,p\_x5\_1,p\_x6\_0,p\_x6\_1,p\_x7\_0,p\_x7\_1=Bayes(data,test)

p\_1=p\_1\_p\*p\_x0\_1\*p\_x1\_1\*p\_x2\_1\*p\_x3\_1\*p\_x4\_1\*p\_x5\_1\*p\_x6\_1\*p\_x7\_1

p\_0=p\_0\_p\*p\_x0\_0\*p\_x1\_0\*p\_x2\_0\*p\_x3\_0\*p\_x4\_0\*p\_x5\_0\*p\_x6\_0\*p\_x7\_0

#看一下结果是否和博客相同

#和博客不同，博客中的好瓜凹陷是6个其实数据中只有五个，数据中好瓜蜷缩是0.375其实应该是0.625

print(p\_1,p\_0)

if p\_1>p\_0:

    print('这是好瓜')

else:

    print('这是坏瓜')

#拉普拉斯修正

#对于某个离散值 如果训练集没有与某个类同时出现过，那么其p(x|c)=0，这会导致连乘时不管其他属性取值，其概率都为0

#采用拉普拉斯修正  p(c)=(Dc+1)/(D+N)  p(c|x)=(Dcxi+1)/(Di+Ni)

test=['青绿','蜷缩','清脆','清晰','凹陷','硬滑',0.697,0.460]

#在好瓜中没有清脆这一类别，因此采用拉普拉斯修正

def Bayes(data,test):

    d={'好瓜':0,'坏瓜':0,'好瓜'+test[0]:0,'坏瓜'+test[0]:0,'好瓜'+test[1]:0,'坏瓜'+test[1]:0,'好瓜'+test[2]:0,'坏瓜'+test[2]:0,

    '好瓜'+test[3]:0,'坏瓜'+test[3]:0,'好瓜'+test[4]:0,'坏瓜'+test[4]:0,'好瓜'+test[5]:0,'坏瓜'+test[5]:0,

    '好瓜密度ave':0,'好瓜密度var':0,'好瓜含糖率ave':0,'好瓜含糖率var':0,'坏瓜密度ave':0,'坏瓜密度var':0,'坏瓜含糖率ave':0,'坏瓜含糖率var':0}

    good\_density=[]

    bad\_density=[]

    good\_sugar=[]

    bad\_sugar=[]

    for item in np.array(data):

        if item[8]=='是':

            d['好瓜']+=1

            good\_density.append(item[6])

            good\_sugar.append(item[7])

            if item[0]==test[0]:

                d['好瓜'+test[0]]+=1

            if item[1]==test[1]:

                d['好瓜'+test[1]]+=1

            if item[2]==test[2]:

                d['好瓜'+test[2]]+=1

            if item[3]==test[3]:

                d['好瓜'+test[3]]+=1

            if item[4]==test[4]:

                d['好瓜'+test[4]]+=1

            if item[5]==test[5]:

                d['好瓜'+test[5]]+=1

        elif item[8]=='否':

            d['坏瓜']+=1

            bad\_density.append(item[6])

            bad\_sugar.append(item[7])

            if item[0]==test[0]:

                d['坏瓜'+test[0]]+=1

            if item[1]==test[1]:

                d['坏瓜'+test[1]]+=1

            if item[2]==test[2]:

                d['坏瓜'+test[2]]+=1

            if item[3]==test[3]:

                d['坏瓜'+test[3]]+=1

            if item[4]==test[4]:

                d['坏瓜'+test[4]]+=1

            if item[5]==test[5]:

                d['坏瓜'+test[5]]+=1

    d['好瓜密度ave']=np.average(good\_density)

    d['好瓜密度var']=np.std(good\_density,ddof=1)

    d['好瓜含糖率ave']=np.average(good\_sugar)

    d['好瓜含糖率var']=np.std(good\_sugar,ddof=1)

    d['坏瓜密度ave']=np.average(bad\_density)

    d['坏瓜密度var']=np.std(bad\_density,ddof=1)

    d['坏瓜含糖率ave']=np.average(bad\_sugar)

    d['坏瓜含糖率var']=np.std(bad\_sugar,ddof=1)

    num=len(np.unique(list(data['好瓜'])))

    num0=len(np.unique(list(data['色泽'])))

    num1=len(np.unique(list(data['根蒂'])))

    num2=len(np.unique(list(data['敲声'])))

    num3=len(np.unique(list(data['纹理'])))

    num4=len(np.unique(list(data['脐部'])))

    num5=len(np.unique(list(data['触感'])))

    p\_1\_p=round((d['好瓜']+1)/(len(data['好瓜'])+num),3)

    p\_0\_p=round((d['坏瓜']+1)/(len(data['好瓜'])+num),3)

    p\_x0\_1=round((d['好瓜'+test[0]]+1)/(d['好瓜']+num0),3)

    p\_x0\_0=round((d['坏瓜'+test[0]]+1)/(d['坏瓜']+num0),3)

    p\_x1\_1=round((d['好瓜'+test[1]]+1)/(d['好瓜']+num1),3)

    p\_x1\_0=round((d['坏瓜'+test[1]]+1)/(d['坏瓜']+num1),3)

    p\_x2\_1=round((d['好瓜'+test[2]]+1)/(d['好瓜']+num2),3)

    p\_x2\_0=round((d['坏瓜'+test[2]]+1)/(d['坏瓜']+num2),3)

    p\_x3\_1=round((d['好瓜'+test[3]]+1)/(d['好瓜']+num3),3)

    p\_x3\_0=round((d['坏瓜'+test[3]]+1)/(d['坏瓜']+num3),3)

    p\_x4\_1=round((d['好瓜'+test[4]]+1)/(d['好瓜']+num4),3)

    p\_x4\_0=round((d['坏瓜'+test[4]]+1)/(d['坏瓜']+num4),3)

    p\_x5\_1=round((d['好瓜'+test[5]]+1)/(d['好瓜']+num5),3)

    p\_x5\_0=round((d['坏瓜'+test[5]]+1)/(d['坏瓜']+num5),3)

    p\_x6\_1=Continuous\_p(test[6],d['好瓜密度ave'],d['好瓜密度var'])

    p\_x6\_0=Continuous\_p(test[6],d['坏瓜密度ave'],d['坏瓜密度var'])

    p\_x7\_1=Continuous\_p(test[7],d['好瓜含糖率ave'],d['好瓜含糖率var'])

    p\_x7\_0=Continuous\_p(test[7],d['坏瓜含糖率ave'],d['坏瓜含糖率var'])

    return p\_0\_p,p\_1\_p,p\_x0\_0,p\_x0\_1,p\_x1\_0,p\_x1\_1,p\_x2\_0,p\_x2\_1,p\_x3\_0,p\_x3\_1,p\_x4\_0,p\_x4\_1,p\_x5\_0,p\_x5\_1,p\_x6\_0,p\_x6\_1,p\_x7\_0,p\_x7\_1

p\_0\_p,p\_1\_p,p\_x0\_0,p\_x0\_1,p\_x1\_0,p\_x1\_1,p\_x2\_0,p\_x2\_1,p\_x3\_0,p\_x3\_1,p\_x4\_0,p\_x4\_1,p\_x5\_0,p\_x5\_1,p\_x6\_0,p\_x6\_1,p\_x7\_0,p\_x7\_1=Bayes(data,test)

p\_1=p\_1\_p\*p\_x0\_1\*p\_x1\_1\*p\_x2\_1\*p\_x3\_1\*p\_x4\_1\*p\_x5\_1\*p\_x6\_1\*p\_x7\_1

p\_0=p\_0\_p\*p\_x0\_0\*p\_x1\_0\*p\_x2\_0\*p\_x3\_0\*p\_x4\_0\*p\_x5\_0\*p\_x6\_0\*p\_x7\_0

#查看p(清脆|好瓜)是否等于博客中修正过的值

#print(p\_x2\_1)

print(p\_1,p\_0)

if p\_1>p\_0:

    print('这是好瓜')

else:

    print('这是坏瓜')