同济大学课程

python数学建模期末大作业

2017— 2018学年第2学期

年级 大三 专业 计算机科学与技术 学号 1553534 姓名 李帅

**请在你的D:\上，建立目录 ： py+你的学号，如学号 150001，则目录为：D:\py1500001**

**将你编写的程序代码和老丛提供的数据文件，全部拷贝到该目录下。**

**程序代码中，读文件的语句中，数据源都必须从该目录读取。**

**将整个文件夹打包上传，作业成绩以正确率、完成时间排序。**

**一、程序**

1. 经过大量统计，获得中国男童在6岁之前，身高的平均值及对应月份的统计数据，存储在文件boy.txt中，文件的第一列为月数，第二列儿童身高（cm），请你分别用多项式拟合（最高到3次项）及生长模型（设，按统计数字的一倍标准差，确定生长的上限K=123.0），对数据进行建模，并计算每个模型的复相关系数，评价哪个模型更好。

代码：

path="D:\\py1553534\\boy.txt"

import numpy as np

import cmath

data=np.loadtxt(path)

#多项式模型

def computecorrelation(x,y):

x\_bar=np.mean(x)

y\_bar=np.mean(y)

SSR=0

Varx=0

Vary=0

for i in range(0,len(x)):

SSR+=(x[i]-x\_bar)\*(y[i]-y\_bar)

Varx+=(x[i]-x\_bar)\*\*2

Vary+=(y[i]-y\_bar)\*\*2

SST=cmath.sqrt(Varx\*Vary)

return SSR/SST

def polyfot(x,y,degree):

result={}

coef=np.polyfit(x,y,degree)#算出各个回归系数

result["polynomial"]=coef.tolist()

p=np.poly1d(coef)#拟合一条线

y\_hat=p(x)

y\_bar=np.mean(y)

SSR=np.sum((y\_hat-y\_bar)\*\*2)

SST=np.sum((y-y\_bar)\*\*2)

result["determination"]=SSR/SST

return result

for i in range(1,4):

model=polyfot(data[:,0],data[:,1],i)

print(model)

#生长模型

k=123.0

X=[]

y=[]

for i in data:

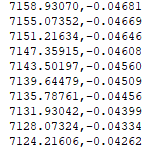
X.append(i[0])

y.append(np.log(k/i[1]-1))

growth=polyfot(X,y,1)

print(growth)

2、某实验中心为客户提供样品检测服务，每个样品单独存储为一个文件，每个文件的格式如下图所示：



可以看出，文件的列间是以逗号间隔的，文件的第一列是采样点的坐标，第二列是仪器的响应。现有某客户的检测样品数据Sam001~Sam005，为建模计算方便，需要将不同样本的第二列，合成一个文件，一列一个样本，请你完成该任务。并将合并后的矩阵，写到将建sam\_merge.txt文件里，列间用\t间隔，每个值占10位字符宽度，保留5位小数。

程序代码放在这里

**import** numpy **as** np  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 ans = []  
 **for** i **in** range(1, 6):  
 file = np.genfromtxt(**"D:\\py1553534\\Sam00{ind}"**.format(ind=i), delimiter=**','**)  
 **if** i == 1:  
 **for** x **in** file[:,1]:  
 ans.append([**"%10.5f"** %x])**else**:  
 l = len(ans)  
 **for** i **in** range(l):  
 ans[i].append((**"%10.5f"** %file[i, 1]))  
  
 ans = np.array(ans)  
 file\_ans = open(**r"D:\py1553534\sam\_merge.txt"**, **"w"**)  
 **for** line **in** ans:  
 print\_line = **"\t"**.join(line)  
 file\_ans.writelines(print\_line)  
 file\_ans.write(**"\n\n"**)

3、针对第二题， 设5个样本的对应的检测目标值（既函数y取值）分别为：

16.579 ， 13.611 ， 14.334 ， 12.865 ， 12.992

请用这5个数据作为y值，与第二题的仪器响应相对应（可以使用老丛给你整理好的sam\_merge1.txt），建立偏最小二乘回归模型（偏最小二乘回归对数据的要求也是一行一个样本）。并对自身进行预测，计算每个样本的预测%误差，其定义为：

abs(真值-预测)/真值\*100。

已知内容：

Sklearn 库中，已经提供了偏最小二乘回归，可以用如下语句引用PLSRegression

from sklearn.cross\_decomposition import PLSRegression

用以下语句构建pls对象。

pls = PLSRegression(n\_components=?,scale=?)

其中参数n\_components是从原始700维空间中提取的主成分数，是一个整数，scale取值为True或False, 取True时，x数据将被按列自标度化，既每列的减其均值，再除以偏差。

建模时，使用如下语句：

pls.fit(x,y)

预报时，采用语句为

yhat=pls.predict(x)

提示：pls总将预测结果用二维矩阵表达。

请你编程，给出代码，分别取2~10个主成分，数据自标或不处理的情况下，最佳的模型方案。既，哪个选择方案误差最小（主成分取多少，是不是自标化等等）？

程序代码：

from sklearn.cross\_decomposition import PLSRegression

import numpy as np

TF = [True, False]

y = np.array([16.579, 13.611, 14.334, 12.865, 12.992])

pls = PLSRegression(n\_components=2,scale=True)

x = np.loadtxt("D:\py1553534\sam\_merge1.txt").T

pls.fit(x, y)

for n\_components\_ind in range(2, 11):

for tf in TF:

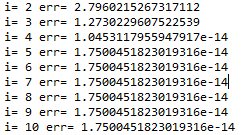
pls = PLSRegression(n\_components=n\_components\_ind,scale=tf)

pls.fit(x, y)

yhat = np.array([val[0] for val in pls.predict(x)])

print(n\_components\_ind, tf, (yhat - y) / y)

pls.predict()

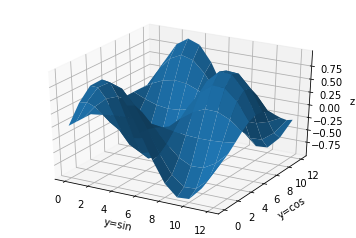
选择对数据进行预处理，你大约将会得到如下的结果（第一列是主成分数，）：

请问：取3主成分和4主成分时，误差值变化非常大，说明什么问题?

取4个主成分后，再增加主成分，没有任何效果，是什么原因造成的？

奇异值问题。

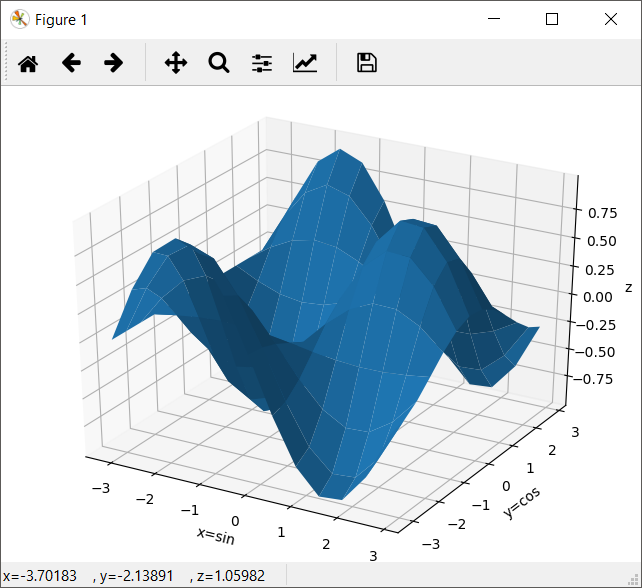
4、制图：一个信号源，在每一瞬间，在x轴方向上，遵循sin曲线，在y轴上，遵循cos曲线，并且，y轴上cos信号的强弱（既信号幅值高低），取决于x轴方向上sin信号在该时刻的强弱。在数学上，x轴的sin形成一个向量， y轴的cos形成另一个向量，观测到的信号，则是这两个向量的外积。请编程产生该信号，并制作3维图形，x、y取值都在[-np.pi,np.pi], 采样间隔0.5。产生的图形如下所示：



制图的程序代码如下

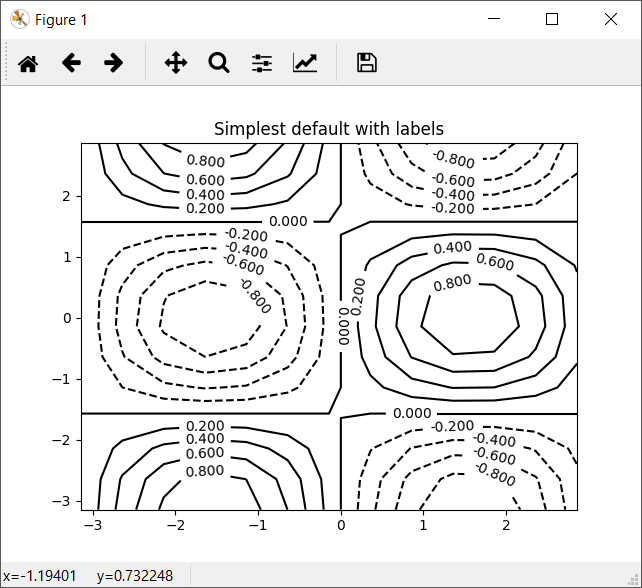
**from** matplotlib **import** pyplot **as** plt  
**import** numpy **as** np  
**from** mpl\_toolkits.mplot3d **import** Axes3D  
  
fig = plt.figure()  
ax = Axes3D(fig)  
X = np.arange(-np.pi,np.pi, 0.5)  
Y = np.arange(-np.pi,np.pi, 0.5)  
  
X, Y = np.meshgrid(X, Y)  
R = np.sqrt(X\*\*2 + Y\*\*2)  
Z = np.sin(X)\*np.cos(Y)  
  
ax.set\_zlabel(**'z'**) *# 坐标轴*ax.set\_ylabel(**'y=cos'**)  
ax.set\_xlabel(**'x=sin'**)

ax.plot\_surface(X, Y, Z, rstride=1, cstride=1)  
  
plt.show()



继续编写求上述图形的等高线图（黑白方式，得到线图横砍10刀）

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** math  
**from** mpl\_toolkits.mplot3d **import** Axes3D  
  
**from** matplotlib **import** pyplot **as** plt  
**import** numpy **as** np  
**from** mpl\_toolkits.mplot3d **import** Axes3D  
  
X = np.arange(-np.pi,np.pi, 0.5)  
Y = np.arange(-np.pi,np.pi, 0.5)  
  
X, Y = np.meshgrid(X, Y)  
R = np.sqrt(X\*\*2 + Y\*\*2)  
Z = np.sin(X)\*np.cos(Y)  
  
  
fig = plt.figure()  
CS = plt.contour(X, Y, Z,10,colors=**"black"**) *# 制作等高线，横砍10刀*plt.clabel(CS, inline=1, fontsize=10) *#inline控制画标签，移除标签下的线*plt.title(**'Simplest default with labels'**)  
plt.show()



5、上述图形对应的矩阵，已经保存到sincos.txt文件中，请使用特征值分析，证明该矩阵的秩为1。

**import** numpy **as** np  
**def** Solve(data):  
 SVD = np.linalg.svd(data, full\_matrices=**False**)  
 T = SVD[0] \* SVD[1]  
 P = SVD[2].T  
 rate = [SVD[1][i] / SVD[1][i + 1] **for** i **in** range(len(SVD[1]) - 1)]  
 k = 0  
 **for** val **in** rate:  
 **if** val > 100:  
 **break  
 else**:  
 k += 1  
  
 T = T[:, :k + 1]  
 P = P[:, :k + 1]  
 **return** k, T, P  
data = np.loadtxt(**r"D:\py1553534\sincos.txt"**).T  
k, T, P = Solve(data)  
print(**"秩为:"**, k + 1)

6 sklearn.datasets中有一套手写数字，共有1797个图片，每个图片大小为8\*8，共被分成10类（10个数字0~9），可以通过如下语句加载：

from sklearn.datasets import load\_digits

digits = load\_digits()

X = digits.data

y = digits.target

X的每一个样本都是64个元素，请参照ppt课件的提供的数字识别技术，使用sklearn中的MLPClassifier建模算法，采用交叉验证技术，20%的样本作为检验样本，并输出**classification\_report**。

提示：y中的每个元素是一个数字，如 1，2等，需要将其转化为正交编码，既1=[0,1,0,0,0,0,0,0,0,0]

*from sklearn.datasets import load\_digits*

*from sklearn.metrics import confusion\_matrix, classification\_report*

*from sklearn.preprocessing import LabelBinarizer*

*from sklearn.model\_selection import train\_test\_split*

*from sklearn.neural\_network import MLPClassifier*

*if \_\_name\_\_ == "\_\_main\_\_":*

*digits = load\_digits()*

*X = digits.data*

*Y = digits.target*

*#Normalization:*

*X -= X.min()*

*X /= X.max()*

*X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2)*

*labels\_train = LabelBinarizer().fit\_transform(Y\_train)*

*labels\_test = LabelBinarizer().fit\_transform(Y\_test)*

*NN = MLPClassifier(hidden\_layer\_sizes=(100, ), alpha=1e-5, solver="adam", activation="relu", max\_iter=10000)*

*print("Start:")*

*NN.fit(X\_train, Y\_train)*

*score = NN.score(X\_test, Y\_test)*

*predictions = NN.predict(X\_test)*

*print(classification\_report(Y\_test, predictions))*

7、请用线性规划求解下列问题

Min x1+x2+x3+x4+x5+x6

s.t.

X1+x6>=60

X1+x2>=70

X2+x3>=60

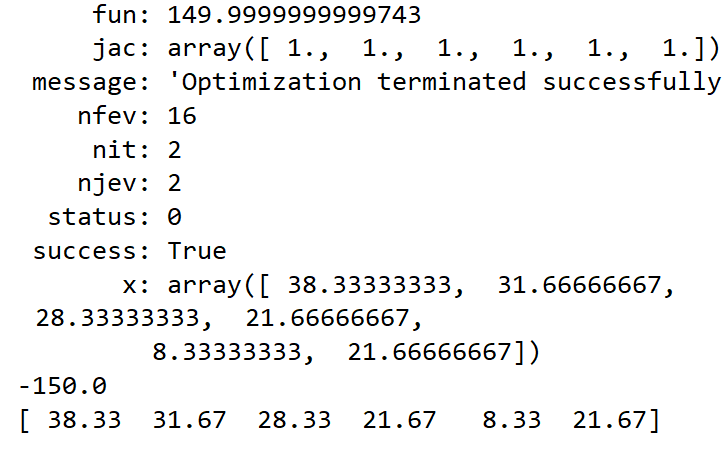
X3+x4>=50

X4+x5>=20

X5+x6>=30

xi>=0

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** math  
  
**import** scipy.optimize **as** so  
**import** numpy **as** np  
  
func = **lambda** x: (x[0] + x[1] + x[2] + x[3] + x[4] + x[5])  
  
func1 = **lambda** x: x[0] + x[5] - 60  
func2 = **lambda** x: x[0] + x[1] - 70  
func3 = **lambda** x: x[1] + x[2] - 60  
func4 = **lambda** x: x[2] + x[3] - 50  
func5 = **lambda** x: x[3] + x[4] - 20  
func6 = **lambda** x: x[4] + x[5] - 30  
  
cons = ({**'type'**: **'ineq'**, **'fun'**: func1},  
 {**'type'**: **'ineq'**, **'fun'**: func2},  
 {**'type'**: **'ineq'**, **'fun'**: func3},  
 {**'type'**: **'ineq'**, **'fun'**: func4},  
 {**'type'**: **'ineq'**, **'fun'**: func5},  
 {**'type'**: **'ineq'**, **'fun'**: func6})  
  
x = [0,0,0,0,0,0]  
  
  
**if** \_\_name\_\_ == **"\_\_main\_\_"**:  
 bnds = ((0, **None**), (0, **None**), (0, **None**), (0, **None**), (0, **None**), (0, **None**))  
 res = so.minimize(func, np.array(x), method=**'SLSQP'**, bounds=bnds, constraints=cons)  
 print(res)  
 print(round(res.fun, 2) \* -1)  
 print(res.x.round(2))



8、已知有下列语句，分为文明语句和粗鲁语句

语句 是否文明语句

my name is laocong. 是

you are very stupid. 否

my boyfriend is SB. 否

you looks very smart, I like you very much. 是

your behavour like a donkey 否

请从上述语句中，找出基本特征词，用朴素贝叶斯建模，并判断下面语句是否是文明语句。

He is a sun boy, full with energy.

path="D:\\py1553534\\"

from sklearn.naive\_bayes import GaussianNB

import jieba

from collections import Counter

import numpy as np

f=open(path+'8.txt').readlines()

data=[]

ans=np.zeros((len(f),1))

for index,line in enumerate(f):

line=line.strip('\n').strip('\t')

line=line.split('<>')

data.append(line[0])

ans[index,0]=float(line[1])

raw\_data=data

al=jieba.lcut(" ".join(data),cut\_all=True)

al=[elem for elem in al if elem!=""]

al=dict(Counter(al))

res=list(reversed(sorted(al.items(),key=lambda x:x[1])))

res=list(filter(lambda x:x[1]>1,res))

for i in res:

print(i)

def formFeature(lines,featureWords): # 根据评语，获得转换矩阵

feature=np.zeros((len(lines),len(featureWords))) #矩阵大小为：用户评价条数\*特征词数

for i in range(feature.shape[0]):

oneLine=lines[i]

info=oneLine.split()

for j in range(feature.shape[1]):

if featureWords[j] in info[0]: #出现特征词，对应位数字=1

feature[i,j]=1.0

return np.array(feature)

#训练语句是1->不文明语句，0->文明语句

bayes=GaussianNB()

print(ans)

featureWords=["stupid","SB","donkey"]

# print(featureWords)

feature=formFeature(raw\_data,featureWords)

model=bayes.fit(feature,ans)

new\_word=["He is a sun boy, full with energy."]

f1=formFeature(new\_word,featureWords)

pred=model.predict(f1)

print("预测结果是：",pred)

#输出是0,所以是文明语句