机器学习导论作业五

161220097 戚赟

2019 年 5 月 18 日

邮箱:986300572@qq.com

1 [20pts] Naive Bayes Classifier

We learned about the naive Bayes classifier using the "property conditional independence hypothesis". Now we have a data set as shown in the following table:

表 1: Dataset

	x_1	x_2	x_3	x_4	y
Instance1	1	1	1	0	1
Instance2	1	1	0	0	0
Instance3	0	0	1	1	0
Instance4	1	0	1	1	1
Instance5	0	0	1	1	1

(1) [10pts] Calculate: $\Pr\{y = 1|\mathbf{x} = (1, 1, 0, 1)\}$ and $\Pr\{y = 0|\mathbf{x} = (1, 1, 0, 1)\}$.

(2) [10pts] After using Laplacian Correction, recalculate the value in

the previous question.

Soulution:

(1)首先计算出先验概率:

$$P(y=1)=\frac{3}{5} \tag{1.1}$$

$$P(y=0)=\frac{2}{5} \tag{1.2}$$

$$\because P(y=1|\mathbf{x}=(1,1,0,1))=\frac{P(y=1\&\mathbf{x}=(1,1,0,1))}{P(\mathbf{x}=(1,1,0,1))} \tag{1.3}$$

$$\therefore P(y=1|\mathbf{x}=(1,1,0,1))=\frac{P(\mathbf{x}=(1,1,0,1)|P(y=1))P(y=1)}{P(\mathbf{x}=(1,1,0,1))} \tag{1.4}$$

下面计算各个的条件概率：

$P(x_1=1|y=1)=\frac{2}{3}$ $P(x_2=1|y=1)=\frac{1}{3}$ $P(x_3=0|y=1)=0$ $P(x_4=1|y=1)=\frac{2}{3}$

$P(x_1=1|y=0)=\frac{1}{2}$ $P(x_2=1|y=0)=\frac{1}{2}$ $P(x_3=0|y=0)=\frac{1}{2}$ $P(x_4=1|y=0)=\frac{1}{2}$

所以计算得出:

$$P(y=1|\mathbf{x}=(1,1,0,1))=\frac{\frac{3}{5}\times\frac{2}{3}\times\frac{1}{3}\times 0\times\frac{2}{3}}{P(\mathbf{x}=(1,1,0,1))}=\frac{0}{P(\mathbf{x}=(1,1,0,1))} \tag{1.5}$$

同理：

$$P(y=0|\mathbf{x}=(1,1,0,1))=\frac{\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2}\times\frac{1}{2}}{P(\mathbf{x}=(1,1,0,1))}=\frac{\frac{1}{40}}{P(\mathbf{x}=(1,1,0,1))} \tag{1.6}$$

而又因为：

$$P(y=1|\mathbf{x}=(1,1,0,1))+P(y=0|\mathbf{x}=(1,1,0,1))=1 \tag{1.7}$$

联立(1.5)(1.6)(1.7)三式，可以解得

$$P(y=1|\mathbf{x}=(1,1,0,1))=0 \tag{1.8}$$

$$P(y=0|\mathbf{x}=(1,1,0,1))=1 \tag{1.9}$$

(2)

先验概率变化为：

$$P(y=1)=\frac{4}{7} \tag{1.10}$$

$$P(y=0)=\frac{3}{7} \tag{1.11}$$

加上拉普拉斯修正之后的条件概率为：

$P(x_1=1|y=1)=\frac{3}{5}$ $P(x_2=1|y=1)=\frac{2}{5}$ $P(x_3=1|y=5)=0$ $P(x_4=1|y=1)=\frac{3}{5}$

$P(x_1=1|y=0)=\frac{1}{2}$ $P(x_2=1|y=0)=\frac{1}{2}$ $P(x_3=0|y=0)=\frac{1}{2}$ $P(x_4=1|y=0)=\frac{1}{2}$

可以解得:

$$P(y=1|\mathbf{x}=(1,1,0,1))=\frac{384}{1009}=0.38057 \tag{1.12}$$

$$P(y=0|\mathbf{x}=(1,1,0,1))=\frac{625}{1099}=0.61943 \tag{1.13}$$

2 [20pts] Bayes Optimal Classifier

For a binary classification task, when data in the two classes satisfies Gauss distribution and have the same variance, please prove that LDA can produce the bayes optimal classifier.

Soulution:

解：最优贝叶斯分类器为:

$$h^*(x)=arg\max_c[P(c|x)]=arg\max_c[\frac{P(c)P(x|c)}{P(x)}] \tag{2.1}$$

由于贝叶斯最优分类，所以$h^*(x)$等价于$h^*(x)=arg\max_c P(c)P(x|c)$

由于符合高斯分布，则设所有类别对应的高斯分布的协方差矩阵相同为

$\sum_c=\sum$

则假设数据维度为d，下面推导过程皆为等价的

$$h^*(x) = arg\max_c P(c)P(x|c) \tag{2.2}$$

$$= arg\max_c \log P(c)P(x|c) \tag{2.3}$$

$$\because P(x|c) = \frac{1}{\sqrt{2\pi}|\Sigma_c|^{\frac{d}{2}}} e^{-\frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c)} \tag{2.4}$$

$$\therefore h^*(x) = arg\max_c[-\frac{1}{2}(x-\mu_c)^T{\Sigma_c}^{-1}(x-\mu_c) + \log P(c)] \tag{2.5}$$

$$= arg\max_c -(x-\mu_c)^T{\Sigma_c}^{-1}(x-\mu_c) + 2\log P(c) \tag{2.6}$$

$$\because -(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c) = 2\mu_c^T\Sigma_c^{-1}x - x^T{\Sigma_c}^{-1}x - \mu_c^T{\Sigma_c}^{-1}\mu_c \tag{2.7}$$

$$\therefore h^*(x) = arg\max_c[2\mu_c^T\Sigma_c^{-1}x - x^T{\Sigma_c}^{-1}x - \mu_c^T{\Sigma_c}^{-1}\mu_c + 2\log P(c)] \tag{2.8}$$

而根据二分类任务，c = 0,1 ，所以只要考虑这两个选最大的，所以做差为

$$(\mu_0-\mu_1)\Sigma^{-1}x - \mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1 + \log\frac{P(1)}{P(0)} \tag{2.9}$$

所以设

$$\mathbf{w}^T = (\mu_0-\mu_1)\Sigma^{-1}, b = -\mu_0^T\Sigma^{-1}\mu_0 - \mu_1^T\Sigma^{-1}\mu_1 + 2\log\frac{P(1)}{P(0)} \tag{2.10}$$

式子(2.9)化为 $\mathbf{w}^Tx + b$，是一条直线。

所以贝叶斯最优分类器是可以通过LDA线性分析获得的。其中的投影变换为$\mathbf{w}^T$和b.

3 [60pts] Ensemble Methods in Practice

Due to their outstanding performance and robustness, ensemble methods are very popular in machine community. In this experiment we will practice ensemble learning methods based on two classic ideas: Boosting and Bagging.

In this experiment, we use an UCI dataset Adult. You can refer to the link[1] to see the data description and download the dataset.

[1] http://archive.ics.uci.edu/ml/datasets/Adult

Adult is an class imbalanced dataset, so we select AUC as the performance measure. You can adopt sklearn to calculate AUC.

(1) [10pts] You need finish the code in Python, and only have two files: AdaBoost.py, RandomForestMain.py. (The training and testing process are implemented in one file for each algorithm.)

(2) [40pts] The is experiment requires to finish the following methods:

- Implement AdaBoost algorithm according to the Fig(8.3), and adopt decision tree as the base learner (For the base learner, you can import sklearn.)
- Implement Random Forest algorithm. Please give a pseudo-code in the experiment report.
- According to the AdaBoost and random forest, analysis the effect of the number of base learners on the performance. Specifically, given the number of base learners, use 5-fold cross validation to obtain the AUC. The range of the number of base learners is decided by yourself.
- Select the best number of base classifiers for AdaBoost and random forests, and obtain the AUC in the test set.

(3) [10pts] In the experimental report, you need to present the detail experimental process. The experimental report needs to be hierarchical and organized, so that the reader can understand the purpose, process and result of the experiment.

Soulution:

(1)我已经按要求实现了程序.使用Python3.7作为编程语言，编写的程序为两个文件 AdaBoost.py, RandomForestMain.py

(2)程序的具体编写

1.Adaboost.py

实现了一个类AdaBoost, 其中的具体函数结构如下

readData(self,filenamse): 读取数据到本地,其中将带有"?"的数据直接清理. filename参数为读取的文件名

clearData(self): 对于读取好的数据进行清理，将string类型按照出现顺序转化为数字，方便计算

Adaboost(self,X,y,num): Adaboost算法,X为特征矩阵,y为结果向量, Num为跑的轮数也就是基学习器的数量,返回树和树的权重

adaboost_predict(self,X, y, trees, trees_weights) : 对于交叉验证的预测,trees为决策树,tree_weights为决策树的权重

adaboost_predict_2(self, X, y, trees, trees_weights) : 对于测试集的预测,trees为决策树,tree_weights为决策树的权重

cross_validation(self,times) :五折交叉验证,times为每轮基学习器的数量

2.RandomForestMain.py

实现了一个类RandomForest,其中的具体函数结构如下

readData(self,filenamse): 读取数据到本地,其中将带有"?"的数据直接清理. filename参数为读取的文件名

clearData(self): 对于读取好的数据进行清理，将string类型按照出现顺序转化为数字，方便计算

BootStrap(self,DataSet): 实现BootStrap采样

ChooseFuture(self,NumOfFuture,NumOfRest):实现特征的选取,每次从NumOfFuture个特征值中选择NumOfRest个特征值进行训练.

randomForest(self,DataSet,num):随机森林进行训练,返回决策树和相应的训练特征

AUC_of_Random(self,Data,trees,FeatureList): 对于交叉验证的预测,Data为测试集合,trees为决策树,FeatureList为每棵树所要用到的特征值

ACURACY_of_Random_test(self, Data, trees, FeatureList): 对于测试集的预测,Data为测试集合,trees为决策树,FeatureList为每棵树所要用到的特征值

其中伪代码如下:

其中BootStrap和ChooseFuture在代码中已经实现,功能已给出.

Algorithm 1: 随机森林算法

Input: A training Set S := (x_1,y_1),(x_2,y_2),...,(x_m,y_m)
B := numbers of trees

Output: RandomForest and FeatureList

1 **Fuction: randomForest(S,B)**

Initialize $Trees = \emptyset, FList = \emptyset$

for $i \in [1, B]$ **do**

/* 获取采样后的矩阵 */

2 BootData = BootStrap(S)

/* 每次随机选取一半的特征 */

3 tempF = ChooseFuture(BootrapData.size,BootrapData.size/2)

/* 更改数据矩阵 */

4 resBootData = BootData[tempF]

/* 训练决策树 */

5 treeT = sklearn.DecisionTress(resBootDatap[特征],resBootDatap[结果])

Trees.add(treeT)

FList.add(tempF)

6 **end**

7 **return Trees,FList**

3.利用交叉验证验证基分类器的数量

1.Adaboost的AUC曲线如下:

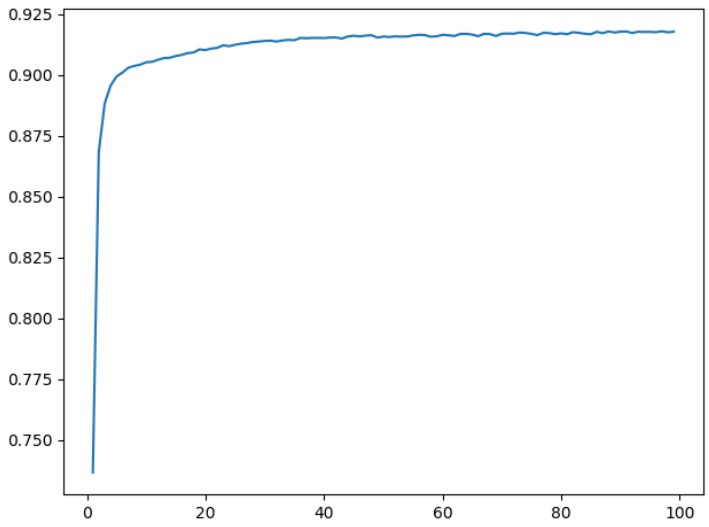

图 1: Adaboost的AUC曲线如下

可以看到两个特点: 1. 随着分类器的数目逐渐上升，AUC的得分也不断的上升，但是后面数量在40之后会逐渐波动。 2.AUC在达到90%左右之后开始摆动，分类器大AUC值不一定好

通过放大查看获得最大分类器数量为45.

2.RandomForest的AUC曲线如下:

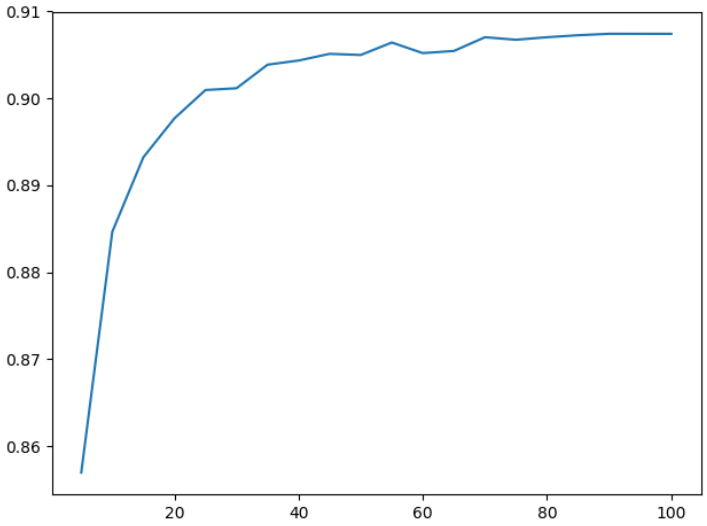

图 2: RandomForest

可见的，虽然中间有波动，但是随着数量器的增加，AUC值是不断上升的，在75的时候稳定在最大值，所以选择基分类器数量为75.

4.验证在测试集上的AUC值和准确率

1.Adaboost的基分类器数量为45时，结果如下:

```
AUC_aver: 0.9153605395347263
准确率: 0.8582337317397079
```

2.RandomForest的基分类器数量为75时，结果如下:

```
测试集AUC为: 0.905360249333841
测试集准确度为: 0.8551128818061089
```