第116天：机器学习算法之朴素贝叶斯理论

原创 某某白米饭 Python技术 1月16日

朴素贝叶斯（Naive Bayesian Mode，NBM）

贝叶斯由来

贝叶斯是由英国学者托马斯·贝叶斯 提出的一种纳推理的理论，后来发展为一种系统的统计推断方法。被称为贝叶斯方法。

朴素贝叶斯

朴素贝叶斯法是基于**贝叶斯定理**与**特征条件独立**假设的分类方法。优点是在数据较少的情况下仍然有效，可以处理多类别的问题。缺点是对于输入数据的装备方式较为敏感。适用于标称型的数据。

特征条件独立：假设 X 的 N 个特征在类确定的条件下都是条件独立的。这样大大简化了计算的复杂度，但是会牺牲一些准确性。

标称型数据：只在有限目标集中取值，比如真与假。

贝叶斯定理

条件概率就是指在事件 B 发生的情况下事件 A 发生的概率，用 P(A|B) 表示，读作 "A 在 B 发生的条件下发生的概率"。

根据文氏图，可以看出在事件 B 发生的情况下，事件 A 发生的概率为 P(A∩B) 除以P(B)。

$$P(A \cap B) = P(A)P(B)$$
$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

所以

$$P(A \cap B) = P(A|B)P(B)$$

同理可得

$$P(A \cap B) = P(B|A)P(A)$$

所以

$$P(A|B)P(B) = P(B|A)P(A)$$

得到

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

其中：

1. P(A) 是 A 的先验概率或边缘概率，不考虑 B 的因素
2. P(A|B) 是已知 B 发生后 A 的条件概率，也称作 A 的后验概率。
3. P(B|A) 是已知 A 发生后 B 的条件概率，也称作 B 的后验概率，称作似然度。
4. P(B) 是 B 的先验概率或边缘概率，称作标准化常量。
5. P(B|A)/P(B) 称作标准似然度。

示例1：桶中的石子

假设现在有 A 桶 和 B 桶两个桶，A 桶里面装有 4 块石子分别2 块黑色的石子和2块灰色的石子，B 桶里面装有 3 块石子分别为 2 块黑色石子和 1 块灰色石子，那么在这两个桶里面取出任意一个石子且都是灰色的，问这个灰色石子在 A 桶中被取出的概率是多少？

假设在 A 桶里面取出石子为事件 A，取出灰色石子为事件 B，在 A 桶中取出灰色石子的事件概率为 P(B|A)，则：P(A) = 4/7，P(B) = 3/7，P(B|A) = 1/2，按照公式：

所以，在两个桶里面取出任意一个石子且为灰色的，这个灰色石子在 A 桶被取出的概率为 2/3

示例2：根据天气情况判断是否出去游玩

在现实中我们经常按天气情况判断是否出去游玩，下面做成一个表格

天气	温度	湿度	风力	结果
多云	热	高	强	是
多云	热	高	强	否
多云	冷	高	弱	否
多云	冷	高	弱	否
多云	冷	低	弱	是
多云	热	低	中	是
小雨	热	高	弱	否
小雨	冷	高	弱	否
小雨	热	低	中	是
小雨	低	低	强	否

现在有个朋友喊你出去游玩，但是天气是多云、温度较冷、湿度较低、风力强，判断一下是否出去游玩。

套用上面朴素贝叶斯公式 P(类别|特征) 为 P(是|多云、冷、低、弱) 和 P(类别|特征) = P(否|多云、冷、低、弱) 的概率。

如果 P(是|多云、冷、低、弱) > P(否|多云、冷、低、弱)，则为出去游玩。如果 P(是|多云、冷、低、弱) < P(否|多云、冷、低、弱)，则为不出去游玩。

由朴素贝叶斯公式可知：

在朴素贝叶斯中，每个特征都是相互独立的，所以可以拆分成为

统计出去游玩的特征概率

下面就可以将特征一个一个统计计算

1.首先我们整理出去玩的样本，结果为是则出去游玩的样本如下，一共有 3 条数据

天气	温度	湿度	风力	结果
多云	热	高	强	是
多云	冷	低	弱	是
多云	热	低	中	是

天气	温度	湿度	风力	结果
小雨	热	低	中	是

P(是) = 4/10 = 2/5

2.当天气为多云出去游玩 P(多云|是) 的样本统计如下：

天气	温度	湿度	风力	结果
多云	热	高	强	是
多云	冷	低	弱	是
多云	热	低	中	是

P(多云|是) = 3/4

3.当温度为冷出去游玩 P(冷|是) 的样本统计如下：

天气	温度	湿度	风力	结果
多云	冷	低	弱	是

P(冷|是) = 1/4

4.当湿度为低出去游玩 P(低|是) 的样本统计如下

天气	温度	湿度	风力	结果
多云	冷	低	弱	是
多云	热	低	中	是
小雨	热	低	中	是

P(低|是) = 3/4

5.当风力为弱出去游玩 P(弱|是) 的样本统计如下

天气	温度	湿度	风力	结果
多云	冷	低	弱	是

P(弱|是) = 1/4

在这里已经统计出了 P(多云|是)、P(冷|是)、P(低|是)、P(弱|是)、P(是) 的概率，下面开始统计 P(多云)、P(冷)、P(低)、P(弱) 的概率

1.天气为多云 P(多云) 的样本统计一共有 6 条，概率则为 6/10。P(多云) = 6/10 = 3/5

2.温度为冷 P(冷) 的样本统计一共有 4 条，概率则为 4/10。P(冷) = 4/10 = 2/5

3.湿度为冷 P(低) 的样本统计一共有 4 条，概率则为 4/10。P(低) = 4/10 = 2/5

4.风力为弱 P(弱) 的样本统计一共有 5 条，概率则为 1/2。P(弱) = 1/2

计算游玩概率

到这里已经统计出了 P(多云)、P(冷)、P(低)、P(弱) 的概率，把所有数值带入公式：

统计不出去游玩的特征概率

在是否出去游玩中计算了多云、冷、低、强的天气情况下出去游玩 P(是|多云、冷、低、弱) 的概率之后，还需要计算同样的天气情况下不出去游玩 P(否|多云、冷、低、弱)的概率，和上面使用同样的方法计算 P(多云|否)、P(冷|否)、P(低|否)、P(弱|否)*P(否) 的概率。

1.统计不出去游玩 P(否) 的概率，P(否) = 6/10 = 3/5

2.统计当天气为多云不出去游玩 P(多云|否) 的样本概率，P(多云|否) = 3/6 = 1/2

3.统计当温度为冷不出去游玩 P(冷|否) 的样本概率，P(冷|否) = 3/6 = 1/2

4.统计当湿度为低不出去游玩 P(低|否) 的样本概率，P(低|否) = 1/6

5.当风力为弱不出去游玩 P(弱|否) 的样本概率，P(弱|否) = 4/6 = 2/3

计算不游玩概率

上面计算了当不出去游玩是天气情况的概率，则把数值带入公式：

概率比较

很显然的结果：(3/4 * 1/4 * 3/4 * 1/4 * 2/5) / (3/5 * 2/5 * 2/5 * 1/2) < (1/2 * 1/2 * 1/6 * 2/3 * 3/5) / (3/5 * 2/5 * 2/5 * 1/2) 所以 P(是|多云、冷、低、弱) < P(否|多云、冷、低、弱)。

Python 实现

在 Python 中借助 pandas 模块和 numpy 模块可以实现计算朴素贝叶斯，在代码中需要做几件事情：

1. 需要选择样本，如：示例2中的天气样本
2. 计算每个类别的概率，这是先验概率
3. 计算每个特征和类别同时发生的概率，这是后验概率
4. 计算条件概率
5. 比较特征出现在类别的概率

```
1
2  import pandas as pd
3  import numpy as np
4
5  class Nbm(object):
6
7      def getSampleSet(self):
8          dataSet = np.array(pd.read_csv('csv文件'))  #将数据转为数组
9          featureData = dataSet[:, 0 : dataSet.shape[1] - 1] #取出特征
10         labels = dataSet[:, dataSet.shape[1] - 1] #取出类别
11         return featureData, labels
12
13
14     def priori(self, labels):
15         # 求出是和否的先验概率
16         labels = list(labels)
17         priori_ny = {}
18         for label in labels:
19             priori_ny[label] = labels.count(label) / float(len(labels)) # P = count(label) / count(l
20         return priori_ny
21
22     def feature_probability(self, priori_ny, features):
23         # 求出特征概率：多云+是，多云+否，冷+是，冷+否同时发生的概率
24         p_feature_ny = {}
25         for ny in priori_ny.keys():
26             ny_index = [i for i, label in enumerate(labels) if label == ny] # 是、否的下标
27             for j in range(len(features)):
28                 f_index = [i for i, feature in enumerate(trainData[:, j]) if feature == features[j]]
29                 xy_count = len(set(f_index) & set(ny_index)) # 类别和特征下标相同的长度
30                 pkey = str(features[j]) + '+' + str(ny)
31                 p_feature_ny[pkey] = xy_count / float(len(labels)) # 特征和类别同时发生的概率
32         return p_feature_ny
33
34     def conditional_probability(self, priori_ny, feature_probability, features):
35         #求出条件概率
36         P = {}
37         for y in priori_ny.keys():
```

```
            for x in features:
                pkey = str(x) + '|' + str(y)
                P[pkey] = feature_probability[str(x) + '+' + str(y)] / float(priori_ny[y])  # P[X1/Y
        return P

    def classify(self, priori_ny, feature_probability, features):


        #求条件概率
        p = self.conditional_probability(priori_ny, feature_probability, features)

        #求出[多云、冷、低、弱]所属类别
        f = {}
        for ny in priori_ny:
            f[ny] = priori_ny[ny]
            for x in features:
                f[ny] = f[ny] * p[str(x)+'|'+str(ny)]   #计算P(多云|是)*P(冷|是)*P(低|是)*P(弱|是)*P

        return max(f, key=f.get)  #概率最大值对应的类别


if __name__ == '__main__':
    nbm = Nbm()
    features = ['多云', '冷', '低', '弱']
    trainData, labels = nbm.getSampleSet()
    priori_ny = nbm.priori(labels)

    feature_probability = nbm.feature_probability(priori_ny, features)

    result = nbm.classify(priori_ny, feature_probability, features)

    print(features, '的结果是', result)
```

总结

简单的介绍了朴素贝叶斯的一些概念，用了两个示例来增强朴素贝叶斯的学习，希望对大家有所帮助。

参考资料

《机器学习实战》

https://baike.baidu.com/item/贝叶斯公式

https://www.ruanyifeng.com/blog/2011/08/bayesian_inference_part_one.html

https://zhuanlan.zhihu.com/p/26262151

代码地址

> 示例代码：https://github.com/JustDoPython/python-100-day/tree/master/day-116

系列文章

第 115 天：Python 到底是值传递还是引用传递
第 114 天：三木板模型算法项目实战
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