第5讲 朴素贝叶斯

主讲教师：欧新宇

February 21, 2020

Outlines

- 朴素贝叶斯的基本概念
- 贝努利贝叶斯
- 高斯贝叶斯
- 多项式贝叶斯
- 贝叶斯实战——肿瘤判断

朴素贝叶斯的基本概念

贝叶斯方法是以**贝叶斯原理**为基础，使用概率统计的知识对样本数据集进行分类。由于其有着坚实的数学基础，在相当的一段时期，贝叶斯分类算法的都具有**较低的误判率**。

贝叶斯方法的特点是结合***先验概率***和***后验概率***，即避免了只使用先验概率的主观偏见，也避免了单独使用样本信息的过拟合现象。贝叶斯分类算法在数据集较大的情况下表现出较高的准确率，同时算法本身也比较简单。

朴素贝叶斯（Naive Bayesian) 算法是一种基于贝叶斯理论的有"**监督学习算法**"，它假定给定目标值的属性之间是相互条件独立(IID)的，因此称之为"朴素"。

关于朴素贝叶斯的简单例子

关于朴素贝叶斯的简单例子

已知：

- $P(A)$: 表示天气预报今日降水的概率
- $P(B)$: 表示晚高峰堵车的概率
- $P(B|A)$: 如果下雨，晚高峰堵车的概率

那么，当堵车时，下雨的概率为：$P(A|B)=\dfrac{P(B|A)\cdot P(A)}{P(B)}$

设，$P(A)=50\%,\ P(B)=80\%,\ P(B|A)=95\%$，

则堵车时，下雨的概率$P(A|B)=\dfrac{P(B|A)\cdot P(A)}{P(B)}=\dfrac{0.95\times 0.5}{0.8}=0.59375$

关于朴素贝叶斯的简单例子

给定若干条件：

- 假设，有三种与天气有关的气象现象，刮北风、闷热和多云。我们可以用布尔数据表示这些气象现象的状态。例如：刮北风 $= 1$，不闷热 $= 0$， 多云 $= 1$， 不多云 $= 0$。我们可以将这些与天气预报有关的气象现象称之为特征。换句话说，在这个例子中，每个样本都有三个特征，可以分别表示为：f_1, f_2, f_3
- 在这些气象现象下，给出对天气的预测 $\hat{y}$，取值同样为布尔类型 $\hat{y} = 0,1$ 。
- 假设实际的天气， $y = [0,1,1,0,1,0,0]$，其中有3天下雨，4天天晴（没有雨）

关于朴素贝叶斯的简单例子

我们可以将这些信息汇总如下表:

	刮北风	闷热	多云	天气预报下雨	实际天气
第一天	0	1	0	1	**0**
第二天	1	1	1	0	**1**
第三天	0	1	1	0	**1**
第四天	0	0	0	1	**0**
第五天	0	1	1	0	**1**
第六天	0	1	0	1	**0**
第七天	1	0	0	1	**0**

$$X = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad y = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

关于朴素贝叶斯的简单例子

统计气候现象、预测的天气与实际天气之间的关系

```
[1]: import numpy as np

#将X，y赋值为 np数组；X表示气候现象和预测的天气；y 表示实际的天气情况
X = np.array([[0, 1, 0, 1],
              [1, 1, 1, 0],
              [0, 1, 1, 0],
              [0, 0, 0, 1],
              [0, 1, 1, 0],
              [0, 1, 0, 1],
              [1, 0, 0, 1]])

y = np.array([0, 1, 1, 0, 1, 0, 0])

#对不同分类计算每个特征为 1的数量，并使用字典类型进行保存 {气候现象: 出现的次数}
counts = {}
for label in np.unique(y):
    counts[label] = X[y == label].sum(axis = 0) # 将多维数组中的数值，按列进行相加

print("下雨的天数:{0}天，天晴(不下雨)的天数:{1}天\n".format(sum(y == 1), sum(y == 0)))
print("下雨与气候的关系:\n {}".format(counts))
```

```
下雨的天数:3天，天晴(不下雨)的天数:4天

下雨与气候的关系:
 {0: array([1, 2, 0, 4]), 1: array([1, 3, 3, 0])}
```

- ✓ 当 y = 0 时(不下雨的4天), 有1天刮北风、2天闷热、0天多云，但是这4天都被预报为**有雨**
- ✓ 当 y = 1 时(下雨的3天), 有1天刮北风、3天闷热、3天多云，但这3天都被预测为**没有雨**

奇葩的结果！！！

关于朴素贝叶斯的简单例子

场景一：假设天气预报为晴朗，但出现了多云的情况。试问真实的天气是什么？我们将该问题符号化后可以得到：

条件： $f_1 = 0, f_2 = 0, f_3 = 1, \hat{y} = 0$

求解： y

[2]:
```python
# 导入贝努利贝叶斯库
from sklearn.naive_bayes import BernoulliNB

# 定义一个贝努利贝叶斯分类器，用于实现数据拟合
clf = BernoulliNB()
clf.fit(X, y)

# 按照题设条件设定新数据
New_Day = [[0, 0, 1, 0]]
pred = clf.predict(New_Day)

# 输出预测结果
if pred == [1]:
    print("要下雨了，快收衣服啦！(y = 1)")
else:
    print("太阳出来了！(y = 0)")
```

```
要下雨了，快收衣服啦！(y = 1)
```

一个关于朴素贝叶斯的简单例子

场景二：假设天气预报为有雨，但出现了刮北风，闷热，云量不多的情况。试问真实的天气是什么？

条件：$f_1 = 1, f_2 = 1, f_3 = 0, \hat{y} = 1$

求解：y

[3]:
```
# 导入贝努利贝叶斯库
from sklearn.naive_bayes import BernoulliNB

# 定义一个贝努利贝叶斯分类器，用于实现数据拟合
clf = BernoulliNB()
clf.fit(X, y)

# 按照题设条件设定新数据
New_Day2 = [[1, 1, 0, 1]]
pred2 = clf.predict(New_Day2)

# 输出预测结果
if pred2 == [1]:
    print("要下雨了，快收衣服啦！(y = 1)")
else:
    print("太阳出来了！(y = 0)")
```

```
太阳出来了！(y = 0)
```

一个关于朴素贝叶斯的简单例子

输出每个选项的预测概率

- **预测结果**
  - 第一天预测结果：要下雨了，快收衣服啦！ $(y=1)$
  - 第二天预测结果：太阳出来了！ $(y=0)$
- **选项的概率**

[7]:
```
prob1 = clf.predict_proba(New_Day)
prob2 = clf.predict_proba(New_Day2)

print("第一天的预测概率为：{}".format(prob1))
print("第二天的预测概率为：{}".format(prob2))
```

```
第一天的预测概率为： [[0.13848881 0.86151119]]
第二天的预测概率为： [[0.92340878 0.07659122]]
```

结论：[预测概率] 和 [预测结果] 基本一致，然而贝叶斯算法对于数值预测并擅长，因此predict_proba()的预测结果仅供参考。

朴素贝叶斯算法的不同实现

- **贝努利Bernoulli朴素贝叶斯 (Ch0503BernoulliNB.py)**
- **高斯Gaussian朴素贝叶斯(Ch0504BernoulliNB.py)**
- **多项式Multinomial朴素贝叶斯 (Ch0505BernoulliNB.py)**
- **朴素贝叶斯实战——肿瘤判断 (Ch0506CaseBreastCancer.py)**

朴素贝叶斯算法的不同实现

贝努利Bernoulli朴素贝叶斯

贝努利分布也称为"二项分布"，或者"0-1分布"。对于随机变量X，如果 X 的取值只能为 0 或 1，即 $X = \{0, 1\}$ 则称随机变量X满足贝努利分布，其相应的概率为：

$$f(x) = \begin{cases} P(x=1) = & p \\ p(x=0) = & 1-p \end{cases}$$

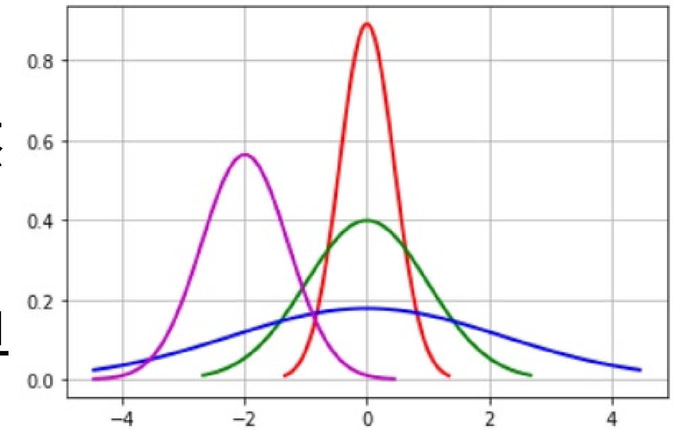
$$s.t. 0 < p < 1$$

贝努利朴素贝叶斯是一种比较**适合于**符合贝努利分布的贝叶斯算法，具体而言就是那些只有两种可能的实验，例如：正面或反面，成功或失败，有缺陷或没有缺陷，病人康复或未康复等。

朴素贝叶斯算法的不同实现

高斯Gaussian朴素贝叶斯

高斯贝叶斯假设样本的特征符合高斯分布，或者说符合正态分布。事实上，自然界的大多数事物都基本满足这个规律。高斯分布（Gaussian distribution），最早由A.棣莫弗在求二项分布的渐近公式中得到。



C.F.高斯在研究测量误差时从另一个角度导出了它。高斯分布是一个在**数学**、**物理**及**工程等领域**都非常重要的概率分布，在**统计学**的许多方面有着重大的影响力。

正态曲线呈钟型，两头低，中间高，左右对称因其曲线呈钟形，因此人们又经常称之为钟形曲线。

朴素贝叶斯算法的不同实现

多项式Multinomial朴素贝叶斯

多项式Multinomial朴素贝叶斯的全称是先验为多项式分布的朴素贝叶斯。MultinomialNB假设特征的**先验概率**为多项式分布，即如下式：

$$P(X_j = x_{jl} | Y = C_k) = \frac{x_{jl} + \lambda}{m_k + n\lambda}$$

其中，$P(X_j=X_{jl} \mid Y=C_k)$ 是第 k 个类别的第 j 维特征的第 l 个取值条件概率。m_k是训练集中输出为第 k 类的样本个数。λ 为一个大于0的常数，通常取值为1，即拉普拉斯平滑，也可以取其他值。

```
[15]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

朴素贝叶斯算法的不同实现

Start：载入数据及数据初始化

[17]:
```
from sklearn.datasets import make_blobs
from sklearn.model_selection import train_test_split

# 生成样本数为 50000, 分类数为 2的数据集, 并按照 75%:25% 的比例进行拆分
X, y = make_blobs(n_samples = 500, centers = 5, random_state = 8)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 8)
```

此处，我们可以轻松地修改make_blobs()函数来实现二分类和多分类的调整。

- 二分类: 超参数 `centers = 2`
- 多分类: 超参数 `centers = 5`， 例如，将样本分为5个类别，可以设置 center = 5

基本流程

1. 数据载入
2. 数据预处理
3. 模型评分（准确率等）
4. 结果可视化

高级操作：

基于学习曲线的参数分析

朴素贝叶斯实战——肿瘤判断

基本步骤

1. 载入数据集
2. 数据集分析
3. 数据预处理（训练集+测试集拆分、正则化等）
4. 基于训练集构建贝叶斯模型
5. 输出模型的准确率评分
6. 对单个样本进行性能评定
7. 对超参数进行分析（ 学习曲线）

欧老师的联系方式

读万卷书 行万里路 只为最好的修炼

QQ：14777591 (宇宙骑士)

Email：ouxinyu@alumni.hust.edu.cn

Tel：18687840023