朴素贝叶斯法

生成模型

张峻伟
2017年10月19日

- **简介：**

朴素贝叶斯分类是贝叶斯分类器的一种，贝叶斯分类算法是统计学的一种分类方法，利用概率统计知识进行分类，其分类原理就是利用**贝叶斯公式**根据某类别的**先验概率和对象特征的在该类别下的条件概率**计算出类别的**后验概率**（即该对象属于某一类的概率），然后选择具有最大后验概率的类作为该对象所属的类

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

条件概率公式 P(B|A)=P(AB)/P(A)

全概率公式 P(X)=∑P(Ai)P(B|Ai)

- **为什么是朴素的**

原因在于对条件概率分布做了条件独立性假设，即假设用于分类的特征在类确定的条件下的条件独立，于是有：

$$P(x|y) = P(a_1|y)P(a_2|y), \ldots, P(a_m|y)$$

对于$P(x|y) = P(a_1, a_2, \ldots a_m|y)$，如果每个特征 a_j 有 S_j 个取值，1<=j<=m，y 的取值有 K 个，那么一共需要考虑的参数个数为$K\prod_{j=1}^{m} S_j$. 特别地，取 S_j=S，那么参数个数为 KS^N，当维数 N 很大的时候，就会发生维数灾难。

现在参数就会大大降低为 KSN（取 S_j=S）,因为$P(a_1|y)$ $P(a_2|y)$ ...和$P(a_m|y)$之间是彼此条件独立的。

$$P(x|y_i)P(y_i) = P(a_1|y_i)P(a_2|y_i), \ldots, P(a_m|y_i)\, P(y_i)$$

$$= P(y_i)\prod_{j=1}^{m} P(a_j|y_i)$$

$$p(y = c_k|x) = \frac{\prod_{i=1}^{M} p(x^i|y = c_k)p(y = c_k)}{\sum_k p(y = c_k)\prod_{i=1}^{M} P(x^i|y = c_k)}$$

$$\frac{p(x|y)p(y)}{}$$

条件概率公式

全概率公式

P(X)=∑P(Ai)P(B|Ai)

对于任何一类y=Ck,分母都是相同的

基本步骤：

（1）首先计算先验概率及条件概率（学习）

$$p(y=c_k)=\frac{\sum_{i=1}^{N} I\,(y_i=c_k)}{N}$$

$$p(x^j=a_{jl}|y=c_k)=\frac{\sum_{i=1}^{N} I(x_i^j=a_{jl},y_i=c_k)}{\sum_{i=1}^{N} I(y_i=c_k)}$$

（2）对于给定的实例，计算该实例属于y=Ck类的概率，确定该实例所属的分类y

$$p(y=c_k|X)=p(y=c_k)\prod_{j=1}^{n} p(X^{(j)}=x^{(j)}|y=c_k)$$

$$y=arg\max_{c_k} p(y=c_k|X)$$

由期望风险函数最小化
得到后验概率最大化

学习过程（参数估计）

- 极大似然估计(模型已知，参数未知)

$$p(x^j = a_{jl}|y = c_k) = \frac{\sum_{i=1}^{N} I(x_i^j = a_{jl}, y_i = c_k)}{\sum_{i=1}^{N} I(y_i = c_k)}$$

可能为0

- 贝叶斯估计 **拉普拉斯平滑**

$$p(y = c_k) = \frac{\sum_{i=1}^{N} I(y_i = c_k) + \lambda}{N + K\lambda}$$

$$p(x^j = a_{jl}|y = c_k) = \frac{\sum_{i=1}^{N} I(x_i^j = a_{jl}, y_i = c_k) + \lambda}{\sum_{i=1}^{N} I(y_i = c_k) + L_j\lambda}$$

课后习题

4.1 用**极大似然估计法**推导朴素贝叶斯法中的先验概率估计公式和条件概率估计公式

随机变量已知，求
参数的概率分布

$$P(Y=c_k)=\frac{\sum_{i=1}^{N} I(y_i=c_k)}{N}$$

$$P(x^{(j)}=a_{jl}|Y=c_k)=\frac{\sum_{i=1}^{N} I(x_i^{(j)}=a_{jl}, y_i=c_k)}{\sum_{i=1}^{N} I(Y=c_k)}$$

$$P(y_1,y_2,\cdots,y_n)=p^{\sum_{i=1}^{N} I(y_i=c_k)}(1-p)^{\sum_{i=1}^{N} I(y_i\neq c_k)}$$

$$\frac{dP(y_1,y_2,\cdots,y_n)}{dp}=\frac{dP(y_1,y_2,\cdots,y_n)}{dp}=p^{\sum_{i=1}^{N} I(y_i=c_k)-1}(1-p)^{\sum_{i=1}^{N} I(y_i\neq c_k)-1}\left((1-p)\sum_{i=1}^{N} I(y_i=c_k)-p\sum_{i=1}^{N} I(y_i\neq c_k)\right)=0$$

$$=(1-P)\sum I(yi=ck)-P\sum I(yi\neq ck)=0$$

$$\sum_{i=1}^{N} I(y_i=c_k)=p\left(\sum_{i=1}^{N} I(y_i=c_k)+\sum_{i=1}^{N} I(y_i\neq c_k)\right) \rightarrow P(Y=c_k)=\frac{\sum_{i=1}^{N} I(y_i=c_k)}{N}$$

同理可得条件概率估计公式

$$P(Y=c_k, x^{(j)}=a_{jl}) = \frac{\sum_{i=1}^{N} I(y_i=c_k, x_i^{(j)}=a_{jl})}{N}$$

$$P(x^{(j)}=a_{jl}|Y=c_k) = \frac{P(Y=c_k, x^{(j)}=a_{jl})}{P(Y=c_k)}$$

$$= \frac{\sum_{i=1}^{N} I(y_i=c_k, x_i^{(j)}=a_{jl})}{N} \Big/ \frac{\sum_{i=1}^{N} I(y_i=c_k)}{N}$$

$$= \frac{\sum_{i=1}^{N} I(y_i=c_k, x_i^{(j)}=a_{jl})}{\sum_{i=1}^{N} I(y_i=c_k)}$$

课后习题

4.2 用贝叶斯估计法推导出朴素贝叶斯法中的概率估计公式

$$P(Y=c_k)=\frac{\sum_{i=1}^{N} I(Y=c_k)+\lambda}{N+K\lambda}$$

$$P\left(x^{(j)}=a_{jl}|Y=c_k\right)=\frac{\sum_{i=1}^{N} I(x^{(j)}=a_{jl},Y=c_k)+\lambda}{\sum_{i=1}^{N} I(Y=c_k)+S_j\lambda}$$

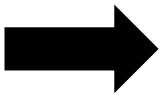
其中λ是参数，K是Y可取的值数，Sj是X(j)可取的值数

加入先验概率，在没有任何信息的情况下可以假设先验概率为均匀概率，即有：

$$p=\frac{1}{K}\Leftrightarrow pK-1=0$$

由上一题可得极大似然下的条件概率约束：

$$pN-\sum_{i=1}^{N} I(y_i=c_k)=0$$



$$\lambda(pK-1)+pN-\sum_{i=1}^{N} I(y_i=c_k)=0$$

$$P(Y=c_k)=\frac{\sum_{i=1}^{N} I(Y=c_k)+\lambda}{N+K\lambda}$$

同理可得：

$$P\left(Y=c_k, x^{(j)}=a_{jl}\right)=\frac{\sum_{i=1}^{N} I\left(y_i=c_k, x_i^{(j)}=a_{jl}\right)+\lambda}{N+KS_j\lambda}$$

$$P\left(x^{(j)}=a_{jl}|Y=c_k\right)=\frac{P\left(Y=c_k, x^{(j)}=a_{jl}\right)}{P(Y=c_k)}$$

$$=\frac{\sum_{i=1}^{N} I\left(y_i=c_k, x_i^{(j)}=a_{jl}\right)+\lambda}{N+KS_j\lambda} \Big/ \frac{\sum_{i=1}^{N} I(y_i=c_k)+\lambda}{N+K\lambda}$$

$$=(\lambda可随便取，所以右边取\lambda=S_j\lambda)$$

$$=\frac{\sum_{i=1}^{N} I\left(y_i=c_k, x_i^{(j)}=a_{jl}\right)+\lambda}{N+KS_j\lambda} \Big/ \frac{\sum_{i=1}^{N} I(y_i=c_k)+S_j\lambda}{N+KS_j\lambda}$$

$$=\frac{\sum_{i=1}^{N} I\left(y_i=c_k, x_i^{(j)}=a_{jl}\right)+\lambda}{\sum_{i=1}^{N} I(y_i=c_k)+S_j\lambda}$$

算法实现

```
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""
import sys
import math
import numpy as np
from collections import Counter

def loadDataSet():#数据格式
    postingList=[['my', 'dog', 'has', 'flea', 'problems', 'help', 'please'],
                 ['maybe', 'not', 'take', 'him', 'to', 'dog', 'park', 'stupid'],
                 ['my', 'dalmation', 'is', 'so', 'cute', 'I', 'love', 'him'],
                 ['stop', 'posting', 'stupid', 'worthless', 'garbage'],
                 ['mr', 'licks', 'ate', 'my', 'steak', 'how', 'to', 'stop', 'him'],
                 ['quit', 'buying', 'worthless', 'dog', 'food', 'stupid']]
    classVec = [0,1,0,1,0,1]#1 侮辱性文字 ， 0 代表正常言论
    return postingList,classVec

def createVocabList(dataSet):#创建词汇表
    vocabSet = set([])
    for document in dataSet:
        vocabSet = vocabSet | set(document) #创建并集
    #print(vocabSet)
    return list(vocabSet)

#词袋模型
def bagOfWord2VecMN(vocabList,inputSet):#根据词汇表，讲句子转化为向量
    returnVec = [0]*len(vocabList)
    for word in inputSet:
        if word in vocabList:
            returnVec[vocabList.index(word)] += 1  #针对一个词在文档中出现不止一次情况
    #print("returnVec + ", returnVec)
    return returnVec

########训练算法（朴素贝叶斯分类器训练函数(此处仅处理两类分类问题)）########
#计算每个类别中的文档数目
#对每篇训练文档:
    #对每个类别:
        #如果词条出现文档中->增加该词条的计数值
        #增加所有词条的计数值
    #对每个类别:
        #对每个词条:
            #将该词条的数目除以总词条数目得到条件概率
    #返回每个类别的条件概率
###############
def trainNB0(trainMatrix,trainCategory):
    '''
     trainMatrix:文档矩阵
     trainCategory:每篇文档类别标签
    '''
    numTrainDocs = len(trainMatrix)  #测试数据数量
    numWords = len(trainMatrix[0])   #单行数据数量
    pAbusive = sum(trainCategory)/float(numTrainDocs)  #先验概率
```

谢　谢