

云南大学软件学院

实 验 报 告

姓名:王子陈 学号:20171050008 专业:电子科学与技术 日期: 2019 年 10 月 9 日 成绩:_____
任课教师: 谢仲文

数据挖掘技术 实验一

一、实验目的

1. 掌握朴素贝叶斯算法。
2. 选择一种编程语言应用朴素贝叶斯算法。

二、实验内容

1. 在一个简单的、虚拟的数据集（训练集）上应用离散型朴素贝叶斯模型。该数据集如下：

帅？	性格好？	身高？	上进？	嫁与否
帅	不好	矮	不上进	不嫁
不帅	好	矮	上进	不嫁
帅	好	矮	上进	嫁
不帅	好	高	上进	嫁
帅	不好	矮	上进	不嫁
不帅	不好	矮	不上进	不嫁
帅	好	高	不上进	嫁
不帅	好	高	上进	嫁
帅	好	高	上进	嫁
不帅	不好	高	上进	嫁
帅	好	矮	不上进	不嫁
帅	好	矮	不上进	不嫁

2. 构建预测数据集，并在数据集上应用离散型朴素贝叶斯模型。

三、实验要求

1. 完成实验内容，源码作为实验报告附件一起打为一个压缩包提供。该压缩包要包含实验报告、代码文件。
2. 关键部分要求有注释，注释量不低于 20%
3. 要求独立完成，不得抄袭代码。

四、关键实验步骤（请粘贴关键步骤、代码、实验结果）

(使用 C 语言)

1. 输入训练集,存储在一个 $n \times 5$ 矩阵里(1 代表是,0 代表否)

帅?	性	身	上	嫁?
1	0	0	0	0
0	1	0	1	0
1	1	0	1	1
0	1	1	1	1
1	0	0	1	0
0	0	0	0	0
1	1	1	0	1
0	1	1	1	1
1	1	1	1	1
0	0	1	1	1
1	1	0	0	0
1	1	0	0	0

2. 统计贝叶斯公式所需的各项概率(根据大数定理,中心极限定理,以频率等于概率)

比如要计算一个男人(不帅,性格不好,身高矮,不上进)该嫁的概率:

$$p(\text{嫁}|\text{不帅、性格不好、身高矮、不上进}) = \frac{p(\text{不帅、性格不好、身高矮、不上进}|\text{嫁}) * p(\text{嫁})}{p(\text{不帅、性格不好、身高矮、不上进})}$$

total_prob:

不帅	性格不好	矮	不上进	不嫁	嫁
$\frac{5}{12}$	$\frac{4}{12}$	$\frac{7}{12}$	$\frac{5}{12}$	$\frac{6}{12}$	$\frac{6}{12}$

好特征 good_prob:

帅	性格好	高	上进	
3/6	5/6	5/6	5/6	嫁
4/6	3/6	0	2/6	不嫁

差特征 notgood_prob:

不帅	性格不好	矮	不上进	
3/6	1/6	1/6	1/6	嫁
2/6	3/6	1	4/6	不嫁

代码部分:

//收集样本数据, 统计并计算 total_probility:

```
for(i=0;i<n;i++)
{
    printf("输入 4 个特征:帅,性格好,长得高,上进,输入 1 或 0:");
    scanf("%f %f %f %f %f",&features[i][0],&features[i][1],&features[i][2],&features[i][3],&features[i][4]);
    if(features[i][0]==1) //features[i][0]存放样本特征
    {
        handsome++;
        total_prob[0]=handsome/n; //n 个样本里帅的概率
    }
    if(1==features[i][1])
    {
        kind++;
        total_prob[1]=kind/n; //n 个样本里性格好的概率
    }
    if(1==features[i][2])
    {
        tall++;
        total_prob[2]=tall/n; // n 个样本里长的高的概率
    }
    if(1==features[i][3])
    {
        posit++;
        total_prob[3]=posit/n; // n 个样本里上进的概率
    }
    if(1==features[i][4])
    {
        marry++;
        total_prob[4]=marry/n; //n 个样本里嫁的概率
    }
}
```

//计算好特征的概率

```
for(j=0;j<4;j++) //按每一列(每一个特征)进行遍历
{
    good_m=0;
    good_nm=0;
    for(i=0;i<n;i++)
    {
        if(features[i][j]==1)//如果是好特征
        { //判断好特征嫁没嫁
            if(features[i][4]==1)//好_嫁
                good_m++;
            else good_nm++;//好_不嫁
        }
    }
    array[0][j]= good_m/marry; // 好_嫁/嫁
    array[1][j]=good_nm/(n-marry); // 好_不嫁/不嫁
}
array[0][4]=marry/n; //第 5 个数存放嫁和不嫁的总概率
array[1][4]=(n-marry)/n;
}
```

```
//计算不好特征的概率(用 1-好特征矩阵的每个元素值)
for(i=0;i<2;i++)
    for(j=0;j<5;j++)
        notgood_prob[i][j]=1-good_prob[i][j];
```

3. 预测：输入一位男生的 4 个特征,用贝叶斯公式计算得到概率,给出结论

先算嫁的概率:

$$\begin{aligned}
 &P(\text{嫁}|\text{不帅},\text{性格不好},\text{身高矮},\text{不上进}) \\
 &= P(\text{不帅}|\text{嫁}) * P(\text{性格不好}|\text{嫁}) * P(\text{身高矮}|\text{嫁}) * P(\text{不上进}|\text{嫁}) * P(\text{嫁}) / P(\text{不帅}) * P(\text{性格不好}) * P(\text{身高矮}) * P(\text{不上进}) \\
 &= (3/6 * 1/6 * 1/6 * 1/6 * 6/12) / (5/12 * 4/12 * 7/12 * 5/12) \\
 &= ((18/15552) / 700) / 20736 \\
 &= 0.034285714
 \end{aligned}$$

代码部分:

```
cons=total_prob[4]; /*计算除条件概率之外的部分  $\frac{P(\text{嫁})}{P(\text{不帅})P(\text{性格不好})P(\text{矮})P(\text{不上进})}$ ,用 P(嫁)除以每一个分母的元素*/
for(j=0;j<4;j++) //对每一个特征作判断
{
    if(1==info[j]) //info[]存储输入的测试数据,aspect[]存储该组特征用到的条件概率
    {
        aspect[j]=good_prob[0][j]; //如果是好特征就从 good_prob 列表里取
        cons/=total_prob[j];
    }
    else //如果是不好的特征就从 notgood_prob 列表里取值
    {
        aspect[j]=notgood_prob[0][j];
        cons/=(1-total_prob[j]);
    }
}
p_marry=(aspect[0])*(aspect[1])*(aspect[2])*(aspect[3])*cons;
```

再算不嫁的概率:

$$\begin{aligned}
 &P(\text{不嫁}|\text{不帅},\text{性格不好},\text{身高矮},\text{不上进}) \\
 &= P(\text{不帅}|\text{不嫁}) * P(\text{性格不好}|\text{不嫁}) * P(\text{身高矮}|\text{不嫁}) * P(\text{不上进}|\text{不嫁}) * P(\text{不嫁}) / P(\text{不帅}) * P(\text{性格不好}) * P(\text{身高矮}) * P(\text{不上进}) \\
 &= (2/6 * 3/6 * 1/4 * 6/6 * 12) / (5/12 * 4/12 * 7/12 * 5/12) \\
 &= 1.645714285714
 \end{aligned}$$

//不嫁的概率 p_notmarry:

```
cons=1-total_prob[4]; //单概率部分用 p(不嫁)除以在分母上每一个概率
for(j=0; j<4; j++)
{
    if(1==info[j])
```

```

    {
        aspect[j]=good_prob[1][j]; //不嫁里面特征好的,从 goog_prob 列表里取数
        cons/=total_prob[j];
    }
    else
    {
        aspect[j]=notgood_prob[1][j]; //不嫁里特征不好的,从 notgood_prob 中取
        cons/=(1-total_prob[j]);
    }
}
p_notmarry=(aspect[0])*(aspect[1])*(aspect[2])*(aspect[3])*cons;

//比较嫁和不嫁哪个概率大
if(p_marry>p_notmarry)
    printf("这个男人可以嫁\n");
else
    printf("这个男人不能嫁\n");

```

4. 实验结果

```

C:\Users\MY\Desktop\Marry\bin\Debug\Marry.exe
输入样本的条数n:12
输入4个特征:帅,性格好,长得高,上进,输入1或0:1 0 0 0 0
输入4个特征:帅,性格好,长得高,上进,输入1或0:0 1 0 1 0
输入4个特征:帅,性格好,长得高,上进,输入1或0:1 1 0 1 1
输入4个特征:帅,性格好,长得高,上进,输入1或0:0 1 1 1 1
输入4个特征:帅,性格好,长得高,上进,输入1或0:1 0 0 1 0
输入4个特征:帅,性格好,长得高,上进,输入1或0:0 0 0 0 0
输入4个特征:帅,性格好,长得高,上进,输入1或0:1 1 1 0 1
输入4个特征:帅,性格好,长得高,上进,输入1或0:0 1 1 1 1
输入4个特征:帅,性格好,长得高,上进,输入1或0:1 1 1 1 1
输入4个特征:帅,性格好,长得高,上进,输入1或0:0 0 1 1 1
输入4个特征:帅,性格好,长得高,上进,输入1或0:1 1 0 0 0
输入4个特征:帅,性格好,长得高,上进,输入1或0:1 1 0 0 0
good_prob array:
0.500000    0.833333    0.833333    0.833333    0.500000
0.666667    0.500000    0.000000    0.333333    0.500000
notgood_prob array:
0.500000    0.166667    0.166667    0.166667    0.500000
0.333333    0.500000    1.000000    0.666667    0.500000
total_prob list:
0.583333    0.666667    0.416667    0.583333    0.500000
请输入特征:(按6退出)
他很帅输入1,否则输入0:0
性格好输入1,否则输入0:0
他长得高输入1,矮输入0:0
他很上进输入1,否则输入0:0

嫁的概率是:0.034286
不嫁的概率是:1.645714
这个男人不能嫁请输入特征:(按6退出)

```

因为实验使用的训练数据集, 存在条件概率为 1 和 0 的情况(即 $P(\text{矮}|\text{不嫁})=1$, $P(\text{高}|\text{不嫁})=0$), 所以用贝叶斯公式会计算出大于 1 和等于 0 的结果, 说明该数据集不适合用朴素贝叶斯算法计算