**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**

1. **实验目的**
2. 掌握朴素贝叶斯算法。
3. 选择一种编程语言应用朴素贝叶斯算法。
4. **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：



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nx5矩阵里(1代表是,0代表否)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 帅? | 性格好? | 身高? | 上进? | 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 |

1. **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:

刀 （ 不 帅 、 性 格 不 好 、 身 高 矮 、 不 上 进 1 嫁 ） *p( 嫁 ） 
刀 （ 嫁 《 不 帅 、 性 格 不 好 、 身 高 矮 、 不 一 # 进 ） ： 
p （ 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** |
|  |  |  |  |  |  |

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[][]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];

1. **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:**

P(嫁|不帅,性格不好,身高矮,不上进)

=P(不帅|嫁)\*P(性格不好|嫁)\*P(身高矮|嫁)\*P(不上进|嫁)\*P(嫁)/P(不帅)\*P(性格不 好)\*P(身高矮)\*P(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

代码部分：

cons=total\_prob[4]; /\*计算除条件概率之外的部分,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:**

P(不嫁|不帅,性格不好,身高矮,不上进)

=P(不帅|不嫁)\*P(性格不好|不嫁)\*P(身高矮|不嫁)\*P(不上进|不嫁)\*P(不嫁)/P(不帅)\*P(性格不好)\*P(身高矮)\*P(不上进)

=(2/6∗3/6∗1∗4/6∗6/12)/(5/12∗4/12∗7/12∗5/12)

=1.645714285714

//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**

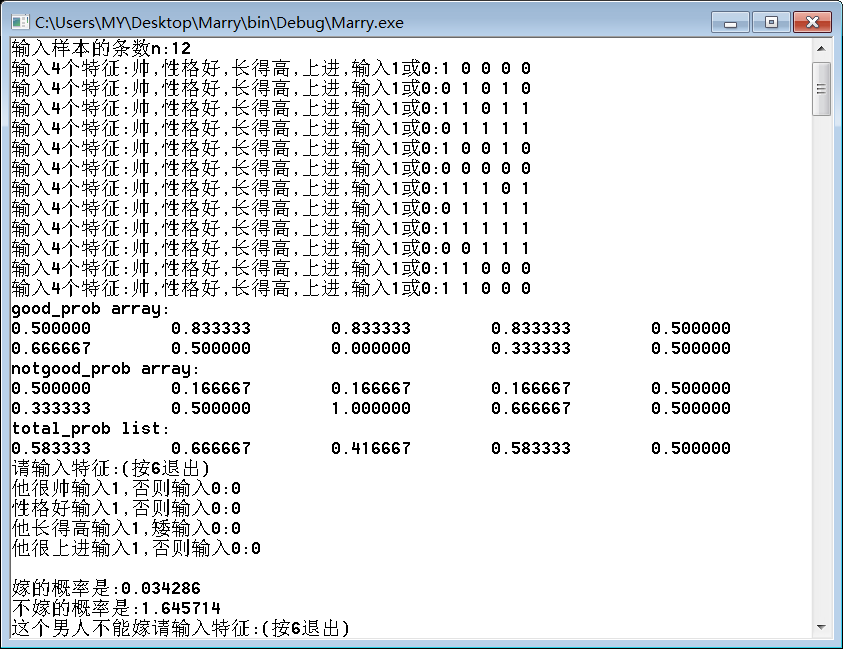
if(p\_marry>p\_notmarry)

printf("这个男人可以嫁\n");

else

printf("这个男人不能嫁\n");

1. **实验结果**



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(矮|不嫁)=1, P(高|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