

云南大学数学与统计学院

上机实践报告

课程名称：信息论基础实验

年级：2013 级

上机实践成绩：

指导教师：陆正福

姓名：金洋

上机实践名称：熵的计算编程实验

学号：20131910023

上机实践日期：

上机实践编号：No. 2

组号：

2016/3/17

上机实践时间：8:35

一、实验目的

给定分布，计算熵；给定原始数据，计算熵

二、实验内容

1. 给定二维分布函数，计算联合熵、条件熵、互信息、各变量的熵。

可选择课本例题 2.2.1 作为程序测试用例。

2. 自行设定原始数据（如一段文本、一幅图像、一个数据表等），按照频率计算符号的分布，进而计算有关的熵。

三、实验环境

1. 个人计算机，任意可以完成实验的平台，如 Java 平台、Python 语言、R 语言、Matlab 平台、Magma 平台等。
2. 对于信息与计算科学专业的学生，建议选择 Java、Python、R 等平台。
3. 对于非信息与计算科学专业的学生，建议选择 Matlab、Magma 等平台。

四、实验记录与实验结果分析

（注意记录实验中遇到的问题。实验报告的评分依据之一是实验记录的细致程度、实验过程的真实性、实验结果的解释和分析。如果涉及实验结果截屏，应选择白底黑字。）

1. 给定二维分布函数，计算联合熵、条件熵、互信息、各变量的熵，按照如下方式求解：

先计算 X, Y 的边缘分布，以此求出 $H(X), H(Y)$ ；

$$\text{条件熵: } H(X|Y) = \sum_{i=1}^m p(Y=i)H(X|Y=i), \quad H(Y|X) = \sum_{j=1}^n p(X=j)H(Y|X=j);$$

$$\begin{aligned} \text{联合熵: } H(X,Y) &= H(X) + H(Y|X) \\ &= H(Y) + H(X|Y) \end{aligned}$$

$$\begin{aligned} \text{互信息: } I(X,Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \end{aligned}$$

给出程序：

Entropy.java

```
package IT2;

public class Entropy {
    protected int m,n;//m*n 分布
    protected float[][] p;
    protected float[] px;
    protected float[] py;

    public Entropy() {

    }

    public Entropy(int m,int n, float[][] p) {
        this.p=new float[m][n];
        this.py=new float[m];
        this.px=new float[n];
        System.arraycopy(p, 0, this.p, 0, p.length);
        this.m=m;
        this.n=n;
    }

    /**
     *
     * @param value 对数的真数
     * @param base 对数的底数
     * @return 对数的值
     */
    public double log(double value, double base) {
        return Math.Log(value) / Math.Log(base);
    }

    /**
     *
     * @param q:一维分布 q
     * @return: 熵
     */
    public float H(float[] q) {
        float h=0;
        int i;
        for (i=0;i<q.length;i++)
            if (q[i]>1e-7)//q[i]=0 时不必相加, 约定  $0\log 0=0$ 
                h+=-q[i]*log(q[i],2);
    }
}
```

```
        return h;
    }

    /**
     *
     * @return H(X)
     */
    public float HX() {
        this.px=new float[n];
        int i,j;
        for (i=0;i<n;i++)
            for (j=0;j<m;j++)
                px[i]+=p[j][i];
        return H(px);
    }

    /**
     *
     * @return H(Y)
     */
    public float HY() {
        this.py=new float[m];
        int i,j;
        for (i=0;i<m;i++)
            for (j=0;j<n;j++)
                py[i]+=p[i][j];
        return H(py);
    }

    /**
     *
     * @return H(X|Y)
     */
    public float HX_Y() {
        float[] px_y=new float[n];
        float hx_y=0;
        int i,j;
        for (i=0;i<m;i++) {
            for (j=0;j<n;j++) px_y[j]=p[i][j]/py[i];
            hx_y+=py[i]*H(px_y);
        }

        return hx_y;
    }

    /**
     *
     * @return H(Y|X)
     */
```

```

public float HY_X() {
    float[] py_x=new float[m];
    float hy_x=0;
    int i,j;
    for (i=0;i<n;i++) {
        for (j=0;j<m;j++) py_x[j]=p[j][i]/px[i];
        hy_x+=px[i]*H(py_x);
    }
    return hy_x;
}

/**
 *
 * @return 联合熵 H(X,Y)
 */
public float HXY() {
    return HX()+HY_X();
}

/**
 *
 * @return 互信息 I(X,Y)
 */
public float IXY() {
    return HX()-HX_Y();
}
}

```

测试主函数:

TestEntropy.java

```

package IT2;
import java.util.Scanner;
public class TestEntropy {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int i,j;
        float[][] p;
        Scanner input=new Scanner(System.in);
        System.out.print("请输入 Y 的取值个数 m: ");
        int m=input.nextInt();
        System.out.print("请输入 X 的取值个数 n: ");
        int n=input.nextInt();
    }
}

```

```

p=new float[m][n];
System.out.println("请输入二维分布 p(y,x)");
for (i=0;i<m;i++)
    for (j=0;j<n;j++)
        p[i][j]=input.nextFloat();

Entropy En=new Entropy(m,n,p);
System.out.println("X 的熵 H(X)=" +En.HX());
System.out.println("Y 的熵 H(Y)=" +En.HY());
System.out.println("条件熵 H(X|Y)=" +En.HX_Y());
System.out.println("条件熵 H(Y|X)=" +En.HY_X());
System.out.println("联合熵 H(X,Y)=" +En.HXY());
System.out.println("互信息 I(X,Y)=" +En.IXY());

}

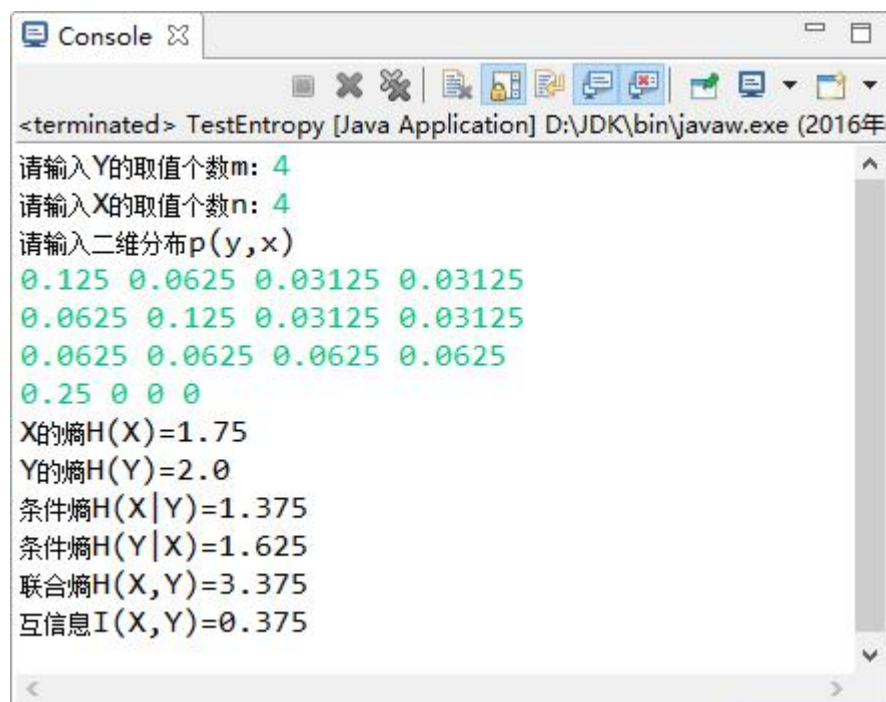
}

```

课本的例题 2.2.1 的分布如下

Y \ X	1	2	3	4
1	0.125	0.0625	0.03125	0.03125
2	0.0625	0.125	0.03125	0.03125
3	0.0625	0.0625	0.0625	0.0625
4	0.25	0	0	0

运行结果：



```

<terminated> TestEntropy [Java Application] D:\JDK\bin\javaw.exe (2016年
请输入Y的取值个数m: 4
请输入X的取值个数n: 4
请输入二维分布p(y,x)
0.125 0.0625 0.03125 0.03125
0.0625 0.125 0.03125 0.03125
0.0625 0.0625 0.0625 0.0625
0.25 0 0 0
X的熵H(X)=1.75
Y的熵H(Y)=2.0
条件熵H(X|Y)=1.375
条件熵H(Y|X)=1.625
联合熵H(X,Y)=3.375
互信息I(X,Y)=0.375

```

2.以输入一段英文小写字母组成的文本为例，输入后统计各字母出现的频率。再利用 1 中 Entropy.java 中的这段代码，便可得到文本的熵

```
/**
 *
 * @param q:一维分布 q
 * @return: 熵
 */
public float H(float[] q) {
    float h=0;
    int i;
    for (i=0;i<q.length;i++)
        if (q[i]>1e-7)//q[i]=0 时不必相加，约定 0log0=0
            h+=-q[i]*log(q[i],2);
    return h;
}
```

主函数程序：

CalcEntropyofData.java

```
package IT2;
import java.io.*;
public class CalcEntropyofData {

    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        int i;
        float[] p=new float[26];
        String text=new String();

        BufferedReader input=new BufferedReader(new
InputStreamReader(System.in));
        System.out.print("请输入一段由英文小写字母组成的文本：");

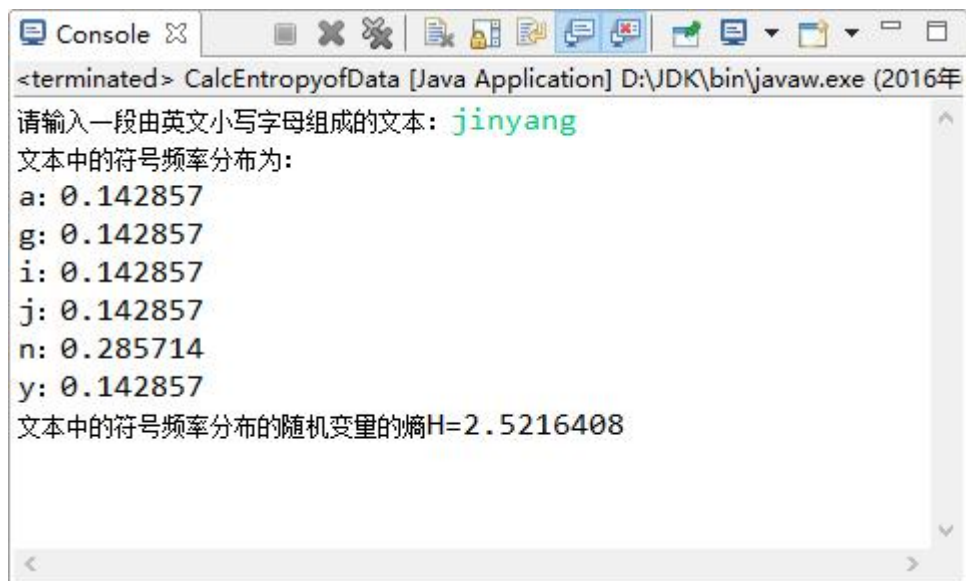
        text=input.readLine();

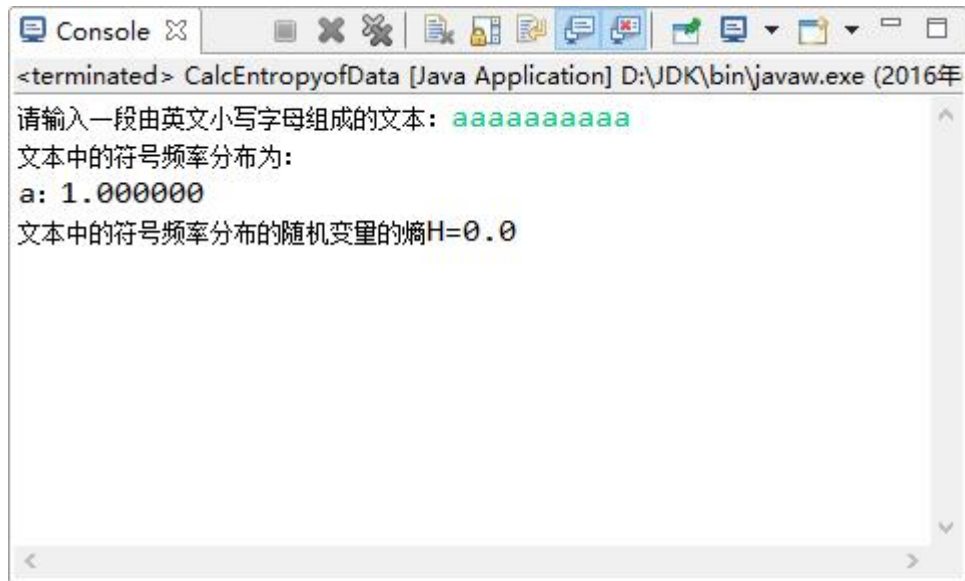
        for (i=0;i<text.length();i++)
            p[text.charAt(i)-97]++;
        System.out.println("文本中的符号频率分布为：");
        for (i=0;i<26;i++)
            if (p[i]>1e-7) {
                p[i]=p[i]/text.length();
                System.out.printf("%c: %f\n",i+97,p[i]);
            }
    }
}
```

```
}
```

```
Entropy En=new Entropy();  
System.out.println("文本中的符号频率分布的随机变量的熵  
H="+En.H(p));  
}  
}
```

运行结果:





五、实验体会

(请认真填写自己的真实体会)

1. 给定二维分布函数，计算联合熵、条件熵、互信息、各变量的熵，可按照如下方式求解：

先计算 X, Y 的边缘分布，以此求出 $H(X), H(Y)$;

$$\text{条件熵: } H(X|Y) = \sum_{i=1}^m p(Y=i)H(X|Y=i), \quad H(Y|X) = \sum_{j=1}^n p(X=j)H(Y|X=j);$$

$$\begin{aligned} \text{联合熵: } H(X,Y) &= H(X) + H(Y|X) \\ &= H(Y) + H(X|Y) \end{aligned}$$

$$\begin{aligned} \text{互信息: } I(X,Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \end{aligned}$$

2. 在已知概率分布，求熵时，要注意对概率为 0 时进行特别处理，一般我们约定 $0\log 0=0$ ， $\log 0$ 在程序里无法计算，要区别对待。否则结果将为 NaN;

六、参考文献

1. (主讲课英文教材)

2. (如有其它参考文献，请列出)