

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

上机实践成绩：

指导教师：陆正福

姓名：金洋

上机实践名称：信息论中常用函数的图形绘制

学号：20131910023

上机实践日期：

2016/3/3

上机实践编号：No.1

组号：

上机实践时间： 8:35

一、实验目的

熟悉信息论中常用函数的图形，为后继学习奠定直观实验基础

二、实验内容

绘制信息论中常用函数的图形

(1) $y = \ln x$

(2) $y = \ln x - x + 1$

(3) $y = x \ln x$

(4) $y = \ln(x)/x$

(5) $y = H(x) = -x \ln x - (1-x) \ln(1-x)$

(6) $D(p \parallel q)(\text{given } q)$

(7) $D(p \parallel q)(\text{given } p)$

(8) $I(X;Y)(\text{given } p(y|x))$

(9) $I(X;Y)(\text{given } p(x))$

三、实验环境

个人计算机，MATLAB 平台

对于信息与计算科学专业的学生，可以尝试选择 Java 平台

对于非信息与计算科学专业的学生，可以选择任意编程平台

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. 对于函数的绘制，MATLAB 功能强大，操作方便，Java 图形绘制优势在于其能够按需要添加多种组件，灵活性较大。

本实验尝试使用 Java 平台绘图，对于函数(1)-(7)可以在 Java 中方便汇出，但是对于函数(8)(9)使用 MATLAB 更为方便，尤其是(9)为多维图像，Java 中绘制并不是明智的选择。

Java 中 JFrame 和 JPanel 的区别：JFrame 是一个窗口，是基于系统产生的，默认情况下，右上角会存在最小化、最大化、关闭按钮和一个 Frame 边框；而 JPanel 是一个面板，是 Swing 提供的一个可以用来画图、放置其他控件的轻量级容器

2. (6) $D(p \parallel q)(\text{given } q)$ 、 (7) $D(p \parallel q)(\text{given } p)$

设 p, q 都是二元分布, $p(0)=1-r, p(1)=r$; 及 $q(0)=1-s, q(1)=s$, 则

$$D(p \parallel q) = (1-r) \log \frac{1-r}{1-s} + r \log \frac{r}{s}$$

3. (1)-(7)函数图像绘制

Jframe.java

package IT1;

import java.awt.*; //抽象窗口工具包, 该包提供了一套与本地图形界面进行交互的接口
import javax.swing.*;

```
public class Jframe extends JFrame {
    private JPanel jp;
    public static void main(String args[]) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Jframe frame=new Jframe();

                    }catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            }
        });
    }

    public Jframe() {
        jp=new JPanel();
        jp.setBackground(Color.WHITE);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        getContentPane().setLayout(new BorderLayout(0,0));
        this.setBackground(Color.WHITE);
        this.setContentPane(jp);
        this.setSize(800,600);
        this.setLocationRelativeTo(null);
        this.setVisible(true);
    }
}
```

JPanel.java

```
package IT1;
```

```
import java.awt.*;
import java.awt.geom.Line2D;
import java.awt.geom.Point2D;
import java.util.Scanner;
```

```
import javax.swing.JComponent;
import javax.swing.JFrame;
```

```
public class JPanel extends JComponent {
    protected int width,height,originX,originY;
    protected static int unit=100;//单位长度包含的像素，也可考虑分别设置 x，
    y 轴的单位长度像素
```

```
    public void paintComponent(Graphics g) {
        width=this.getWidth();
        height=this.getHeight();
        /*新坐标系原点的位置*/
        originX=width/2;
        originY=height/2;

        g.setColor(Color.BLACK);
        this.paintAxis(g);
        this.paintMethod(g);
    }
```

```
    /*
     * 画坐标轴&刻度,可以通过 originX,originY 的处置来改变原点位置;
     */
    public void paintAxis(Graphics g){
        g.setColor(Color.BLACK);
        /*drawLine(int x1, int y1, int x2, int y2) :x1,y1 为起点的坐
        标,x2,y2 为终点的坐标(原画板中的坐标) */
        g.drawLine(0, originY, width, originY);//x 坐标轴
        g.drawLine(originX, 0, originX, height);//y 坐标轴
        g.drawString("0",originX + 2,originY +12); //画原点数字

        /*
         * 画刻度
         * 注意到 y 方向上，在图形界面上越往下像素越大，但是在坐标系中越往上 y
        坐标越大
```

```

    */

    /*从原点开始, x 轴上单位刻度为 unit 像素*/
    for(int i = 1; i*unit <= width/2; i++){
        g.drawLine(originX + i*unit, originY - 10, originX +
i*unit, originY); //x 正向刻度线
        g.drawLine(originX - i*unit, originY - 10, originX -
i*unit, originY); //x 负向刻度线
        g.drawString(String.valueOf(i), originX + i*unit -3,
originY + 12); // x 轴刻度数字
        g.drawString(String.valueOf(i * -1), originX - i*unit -8,
originY + 12); // x 轴刻度数字
    }

    /*从原点开始, y 轴上单位刻度为 unit 像素*/
    for(int i = 1; i*unit <= height/2; i++){

        g.drawLine(originX , originY + i*unit, originX + 10,
originY + i*unit); //y 负向刻度线
        g.drawLine(originX , originY - i*unit, originX + 10,
originY - i*unit); //y 正向刻度线
        g.drawString(String.valueOf(i), originX -12 , originY -
i*unit + 5); // y 轴刻度数字
        g.drawString(String.valueOf(i * -1), originX -12 ,
originY + i*unit + 5); // y 轴刻度数字
    }
}

/**
 *
 * @param x 建立的坐标系中 x 坐标值
 * @return 该坐标值在原画板中的坐标
 */
public double alterX(double x){
    return x+originX;
}

/**
 *
 * @param y 建立的坐标系中 y 坐标值
 * @return 该坐标值在原画板中的坐标
 */
public double alterY(double y) {
    return -1*(y-originY);
}

```

```

public double log(double value, double base) {
    return Math.Log(value) / Math.Log(base);
}

public void paintMethod(Graphics g1) {
    int option=0;
    double s=0,r=0;

    //System.out.println(width);
    //System.out.println(height);
    System.out.println("(1) y=lnx");
    System.out.println("(2) y=lnx-x+1");
    System.out.println("(3) y=xlnx");
    System.out.println("(4) y=ln(x)/x");
    System.out.println("(5) y=H(x)=-xlnx-(1-x)ln(1-x)");
    System.out.println("(6) D(p||q)(given q)");
    System.out.println("(7) D(p||q)(given p)");
    System.out.println("(8) I(X;Y)(given p(y|x))");
    System.out.println("(9) I(X;Y)(given p(x))");
    System.out.println("Please choose a funtion to draw:");

    Scanner input=new Scanner(System.in);
    option=input.nextInt();

    if (option<1 || option>9) {
        System.out.printf("Input error.\n");
        System.exit(0);
    }
    switch (option){
        case 6:
            System.out.printf("设 p,q 都为二值分布, 请给出 q 的分布 s  
(q(1)=s)\n");
            s=input.nextDouble();
            break;
        case 7:
            System.out.printf("设 p,q 都为二值分布, 请给出 p 的分布 r  
(p(1)=r)\n");
            r=input.nextDouble();
            break;
        case 8:
            System.out.printf("请使用对应的 m 脚本文件进行 MATLAB 绘  
图");
            break;
        case 9:
            System.out.printf("请使用对应的 m 脚本文件进行 MATLAB 绘  
图");
            break;
    }
}

```

```

    }

    Point2D temp1,temp2;
    double x = 0,y=0;
    Graphics2D g = (Graphics2D)g1;
    g.setColor(Color.BLUE);

    temp1 = new Point2D.Double(this.alterX(x * unit),
this.alterY(y * unit));
    for(int i = 1 ; i <= width/2; i++){
        x += 1.0/unit;
        y = 0 ;
        switch (option) {
            case 1:y=Math.Log(x);break;
            case 2:y=Math.Log(x)-x+1;break;
            case 3:y=x*Math.Log(x);break;
            case 4:y=Math.Log(x)/x;break;
            case 5:y=-x*Math.Log(x)-(1-x)*Math.Log(1-x);break;
            case 6:
                if (x<0 || x>1) continue;
                else y=(1-x)*log((1-x)/(1-s),2)+x*log(x/s,2);
                break;
            case 7:
                if (x<0 || x>1) continue;
                else y=(1-r)*log((1-r)/(1-x),2)+r*log(r/x,2);
                break;
        }

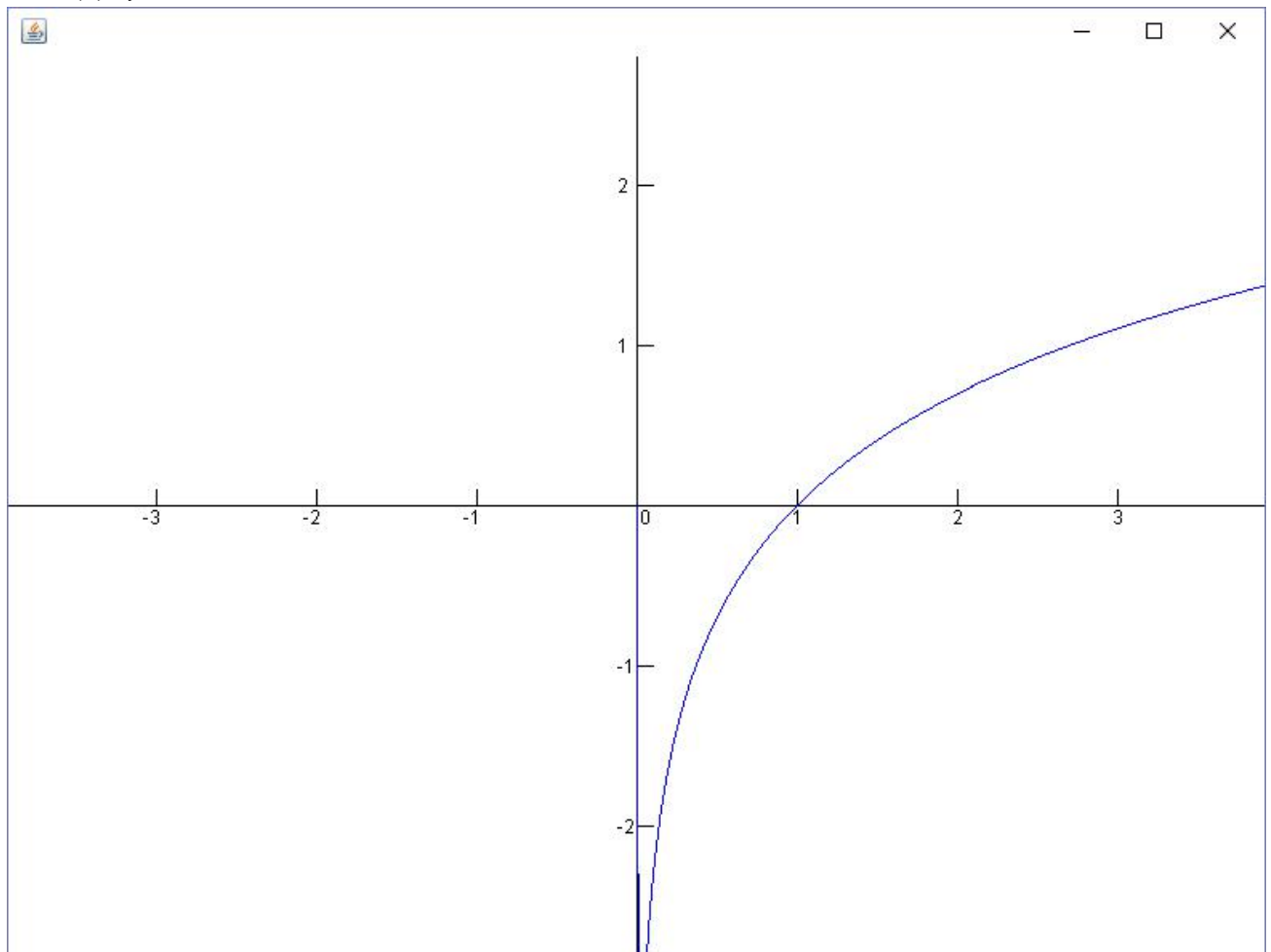
        //图像还在坐标系内
        if ( Math.abs(y) < originY){
            temp2 = new Point2D.Double(this.alterX(x * unit),
this.alterY(y * unit));
            g.draw(new Line2D.Double(temp1, temp2));
            temp1 = temp2;
        }
    }
}
}
}

```

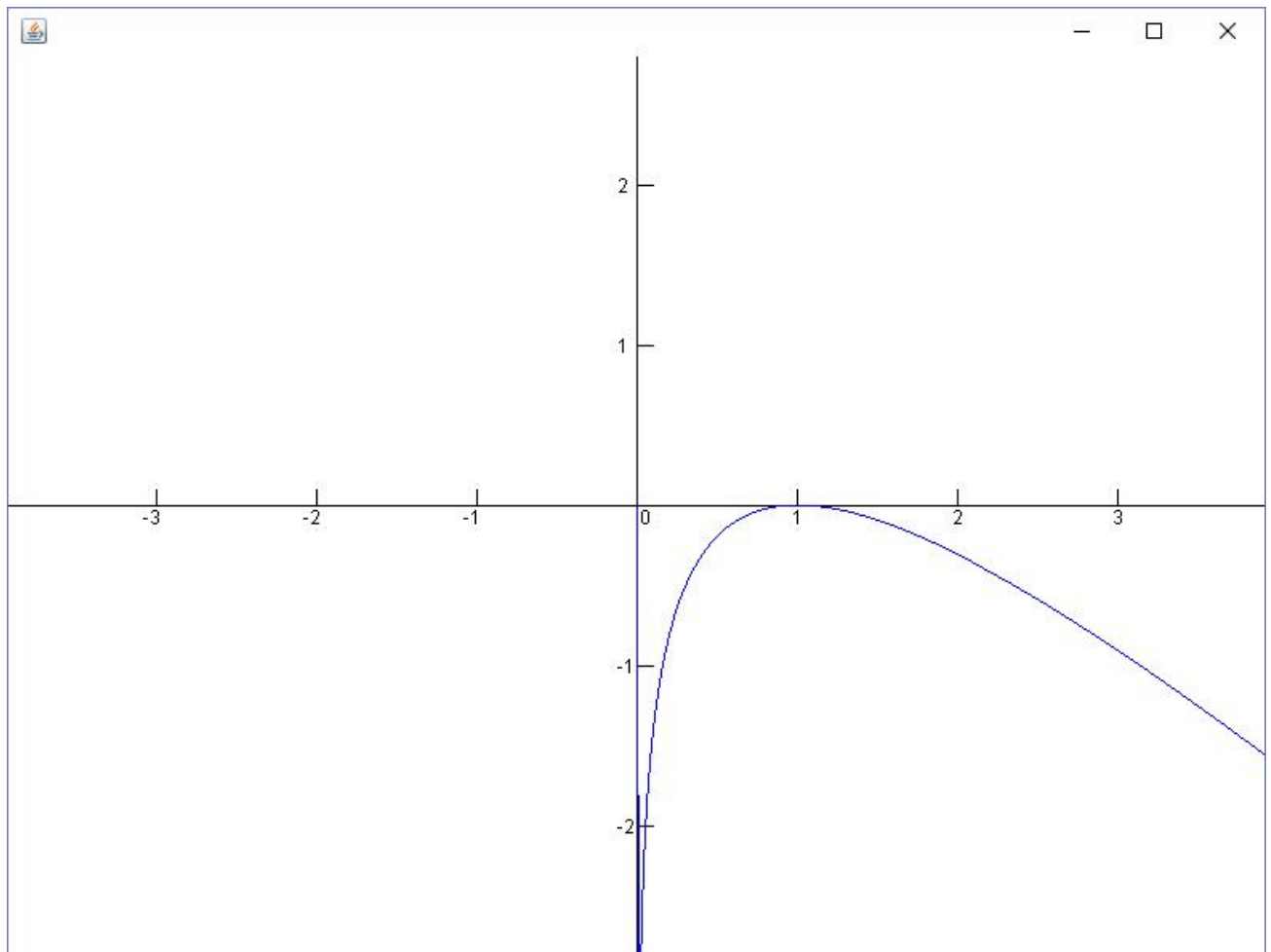
运行结果:

```
Console
Jframe [Java Application] D:\JDK\bin\javaw.exe (2016年5月2日 下午5:
(2)  $y = \ln x - x + 1$ 
(3)  $y = x \ln x$ 
(4)  $y = \ln(x)/x$ 
(5)  $y = H(x) = -x \ln x - (1-x) \ln(1-x)$ 
(6)  $D(p\|q)$  (given  $q$ )
(7)  $D(p\|q)$  (given  $p$ )
(8)  $I(X;Y)$  (given  $p(y|x)$ )
(9)  $I(X;Y)$  (given  $p(x)$ )
Please choose a function to draw:
|
```

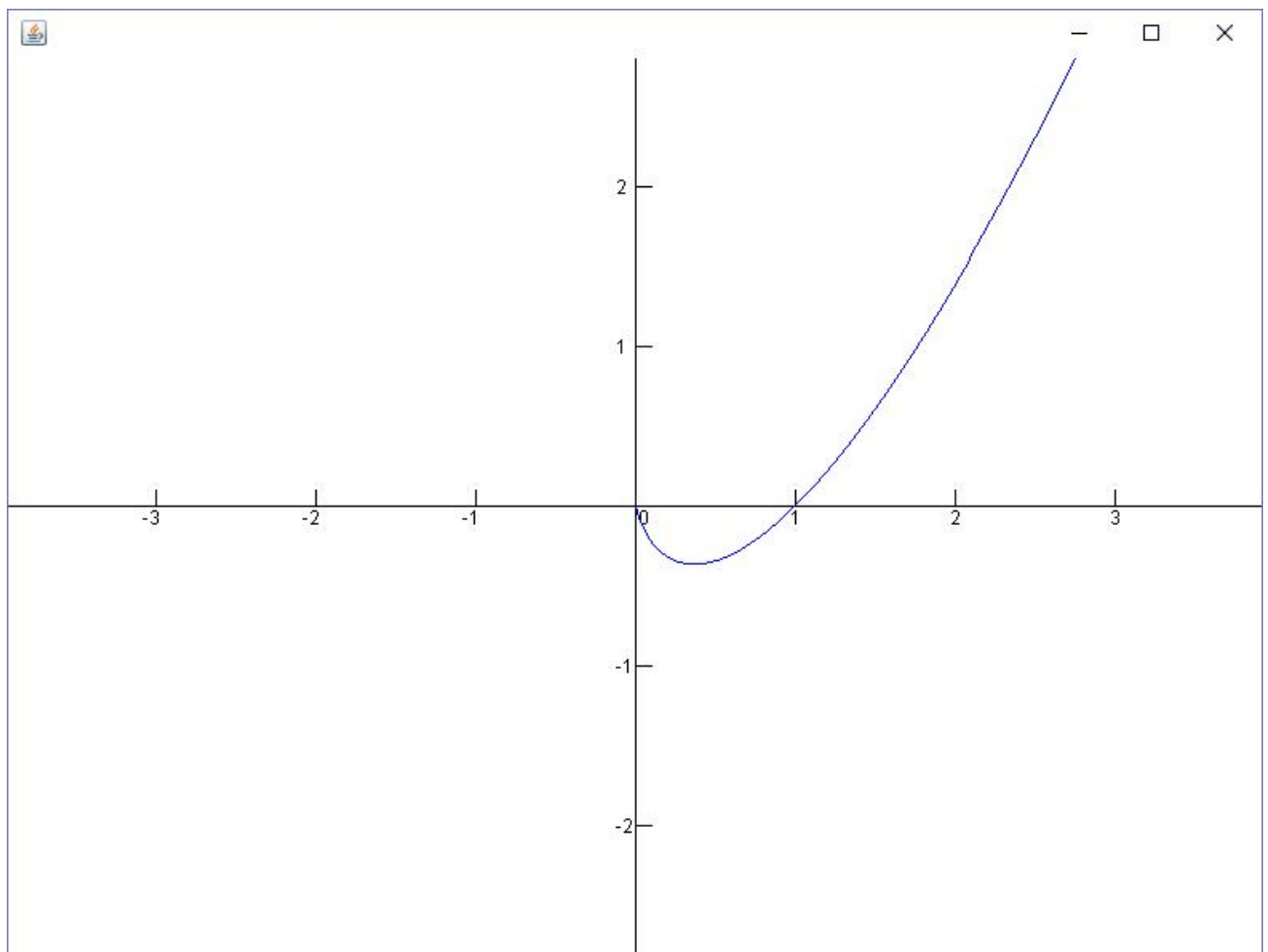
(1) $y = \ln x$



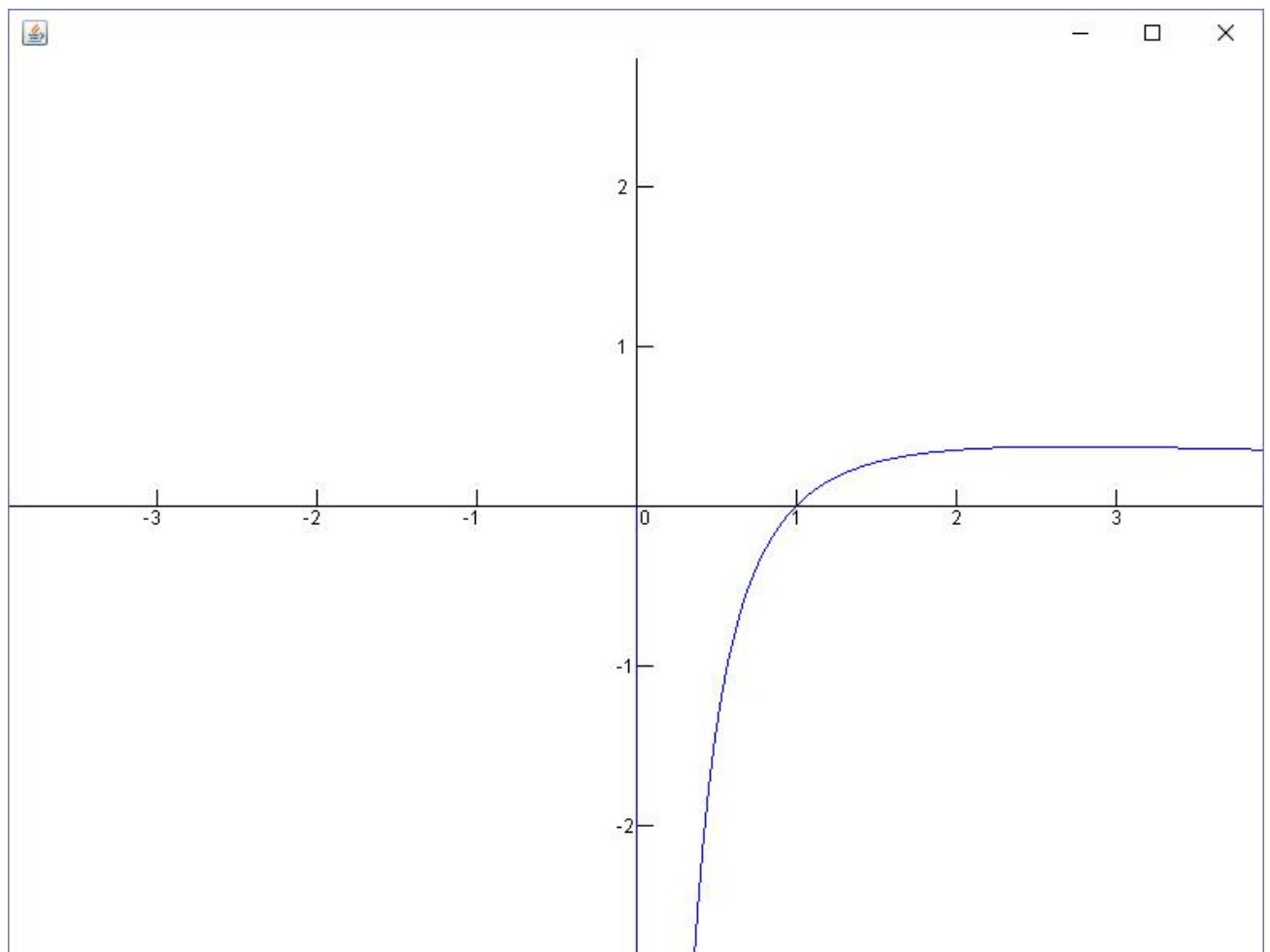
(2) $y = \ln x - x + 1$



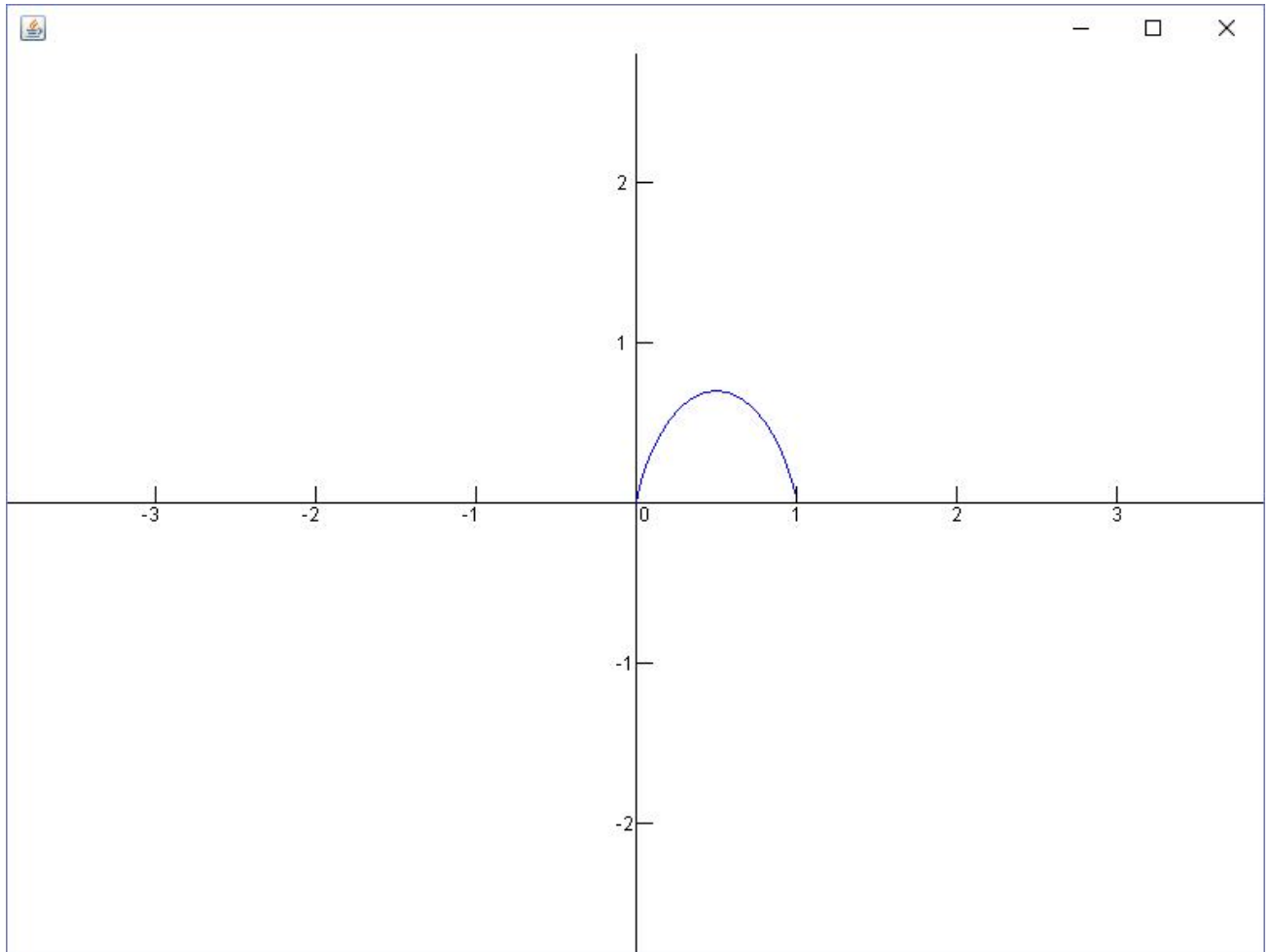
(3) $y = x \ln x$



(4) $y = \ln(x)/x$



(5) $y = H(x) = -x \ln x - (1-x) \ln(1-x)$

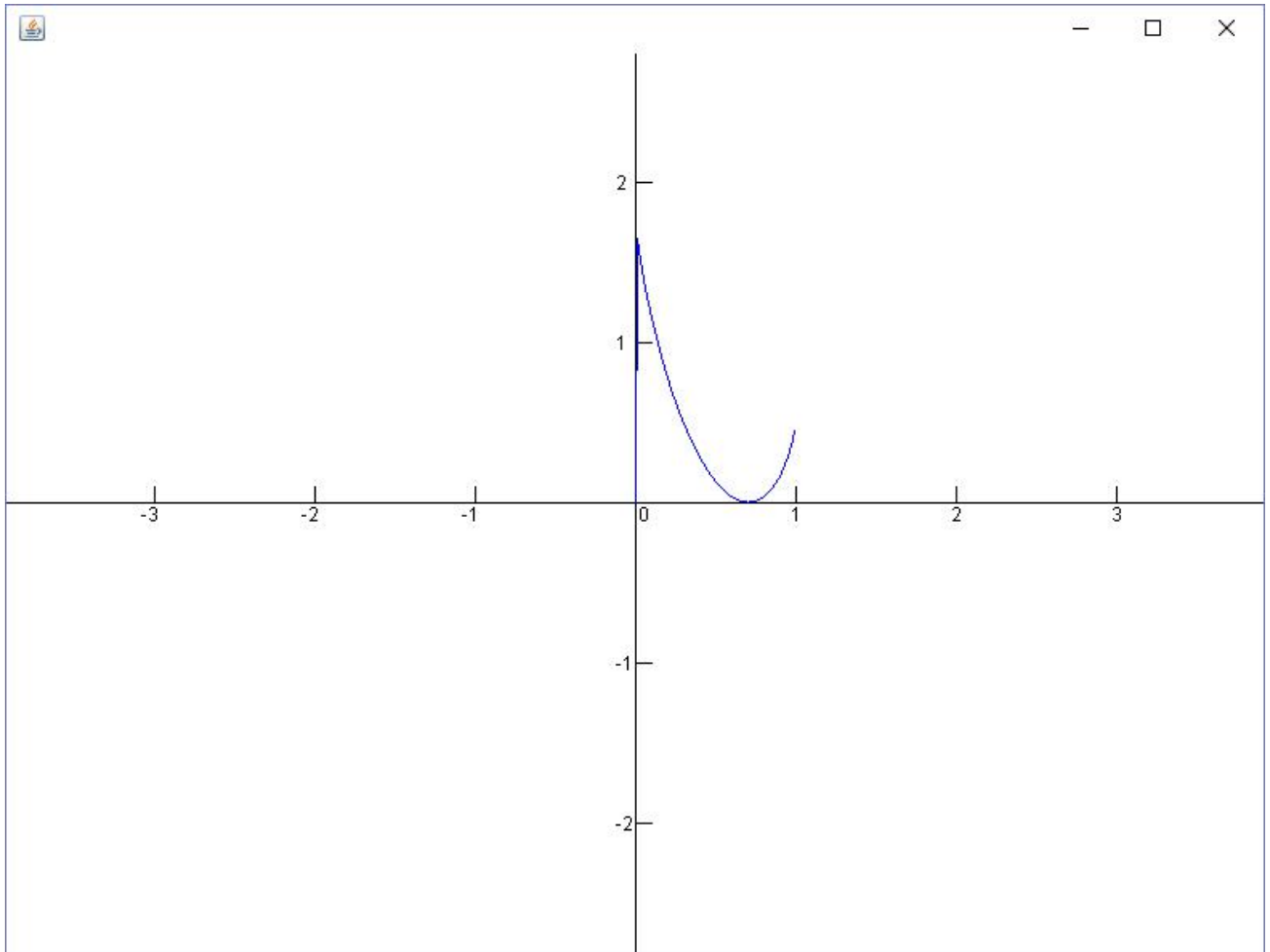


(6) $D(p \parallel q)(\text{given } q)$

```

Console
Jframe [Java Application] D:\JDK\bin\javaw.exe (2016年5月2日 下午5:
(5)  $y=H(x)=-x\ln x-(1-x)\ln(1-x)$ 
(6)  $D(p \parallel q)(\text{given } q)$ 
(7)  $D(p \parallel q)(\text{given } p)$ 
(8)  $I(X;Y)(\text{given } p(y|x))$ 
(9)  $I(X;Y)(\text{given } p(x))$ 
Please choose a funtion to draw:
6
设p,q都为二值分布, 请给出q的分布s ( $q(1)=s$ )
0.7

```

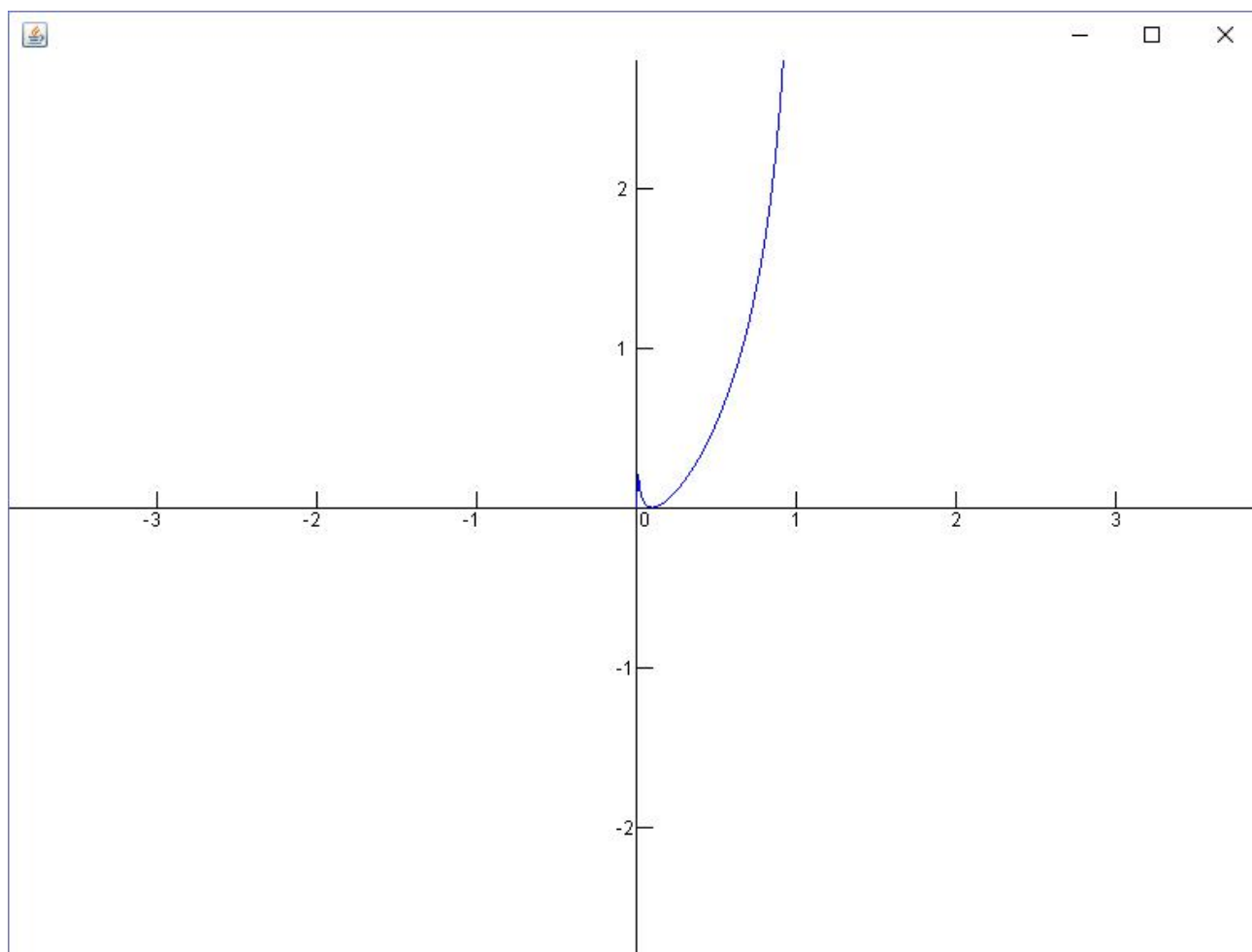


(7) $D(p \parallel q)$ (given p)

```

Console
Jframe [Java Application] D:\JDK\bin\javaw.exe (2016年5月2日 下午5:
(3)  $y = x \ln x$ 
(4)  $y = \ln(x)/x$ 
(5)  $y = H(x) = -x \ln x - (1-x) \ln(1-x)$ 
(6)  $D(p \parallel q)$ (given  $q$ )
(7)  $D(p \parallel q)$ (given  $p$ )
(8)  $I(X;Y)$ (given  $p(y|x)$ )
(9)  $I(X;Y)$ (given  $p(x)$ )
Please choose a funtion to draw:
7
设 $p, q$ 都为二值分布, 请给出 $p$ 的分布 $r$  ( $p(1)=r$ )
0.1

```



4. (8) $I(X;Y)$ (given $p(y|x)$)

$$\begin{aligned}
 I(X;Y) &= H(Y) - H(Y|X) \\
 &= -\sum_y p(y) \log p(y) + \sum_{x,y} p(x,y) \log p(y|x)
 \end{aligned}$$

已知了 $p(y|x)$, 我们以 $p(x)$ 为 $I(X;Y)$ 的自变量

\therefore 已知 $p(y|x)$

\therefore 可计算 $p(x,y) = p(y|x)p(x)$

由全概率公式 $p(y) = \sum_i p(x_i)p(y|x_i)$

所以可计算出 $I(X;Y)$.

具体地, 设 X, Y 都为二值分布, 已知的 $P(Y|X)$ 如下

$P(Y X)$	$Y=1$	$Y=0$
$X=1$	a_{11}	a_{10}
$X=0$	a_{01}	a_{00}

设自变量 $P(X=1)=r, P(X=0)=1-r$, 则

$$P(Y=1)=a_{11}r+a_{01}(1-r),$$

$$P(Y=0)=a_{10}r+a_{00}(1-r);$$

$$P(X=1, Y=1)=a_{11}r,$$

$$P(X=1, Y=0)=a_{10}r,$$

$$P(X=0, Y=1)=a_{01}(1-r),$$

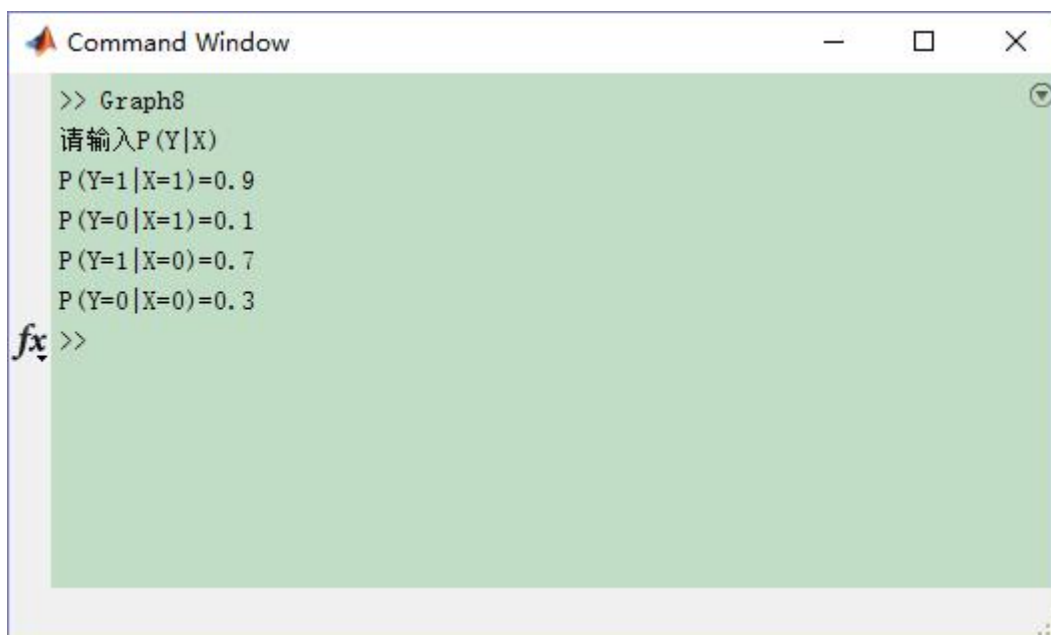
$$P(X=0, Y=0)=a_{00}(1-r)$$

Graph8.m

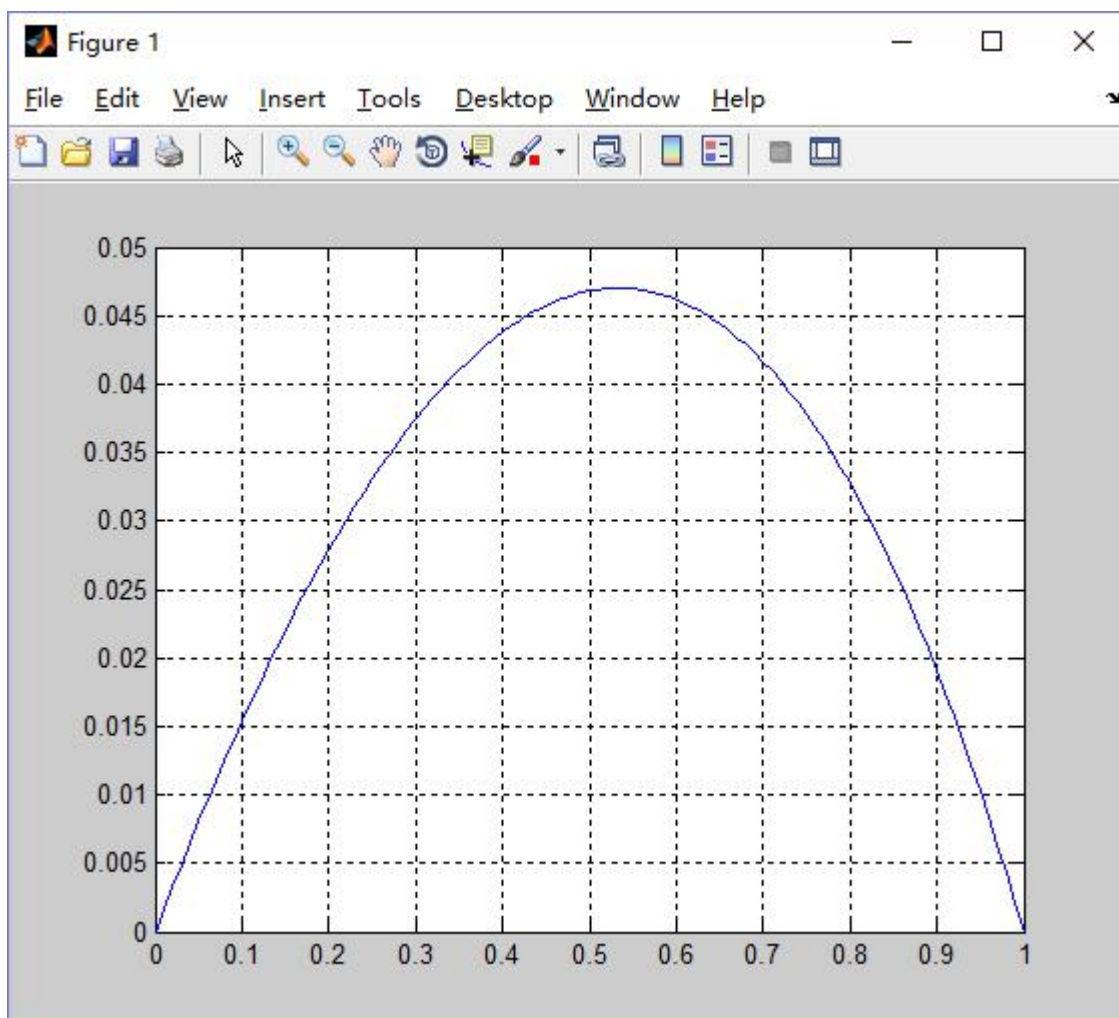
```
fprintf('请输入 P(Y|X) \n');
P(1,1)=input('P(Y=1|X=1)=');
P(1,2)=input('P(Y=0|X=1)=');
P(2,1)=input('P(Y=1|X=0)=');
P(2,2)=input('P(Y=0|X=0)=');
IXY=[];
for r=0:0.01:1
    PY=[r 1-r]*P;
    HY=-sum(PY.*log2(PY));

    PXY=[r r;1-r 1-r].*P;
    HY_X=-sum(sum(PXY.*log2(P)));
    IXY=[IXY, HY-HY_X];
end
r=0:0.01:1;
plot(r, IXY);
grid on;
```

运行结果:



```
>> Graph8
请输入P(Y|X)
P(Y=1|X=1)=0.9
P(Y=0|X=1)=0.1
P(Y=1|X=0)=0.7
P(Y=0|X=0)=0.3
fx >>
```



5. (9) $I(X;Y)$ (given $p(x)$)

$$I(X;Y)$$

$$= H(X) - H(X|Y)$$

$$= -\sum_x p(x) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y)$$

已知了 $p(x)$, 我们以 $p(y|x)$ 为 $I(X;Y)$ 的自变量

$$\text{则 } p(x,y) = p(y|x)p(x);$$

$$\text{由全概率公式 } p(y) = \sum_i p(x_i)p(y|x_i)$$

$$\text{从而 } p(x|y) = p(x,y)/p(y);$$

所以可计算 $I(X;Y)$.

具体地, 设 X, Y 都为二值分布, 已知 $P(X=1)=r, P(X=0)=1-r$, 设两个自变量 t, s ,

$P(Y X)$	$Y=1$	$Y=0$
$X=1$	t	$1-t$
$X=0$	s	$1-s$

Graph9.m

```
r=input('P(X=1)=');
```

```
IXY=[];
```

```
t=eps:0.01:1-eps;
```

```
s=t;
```

```
[T,S]=meshgrid(t,s);
```

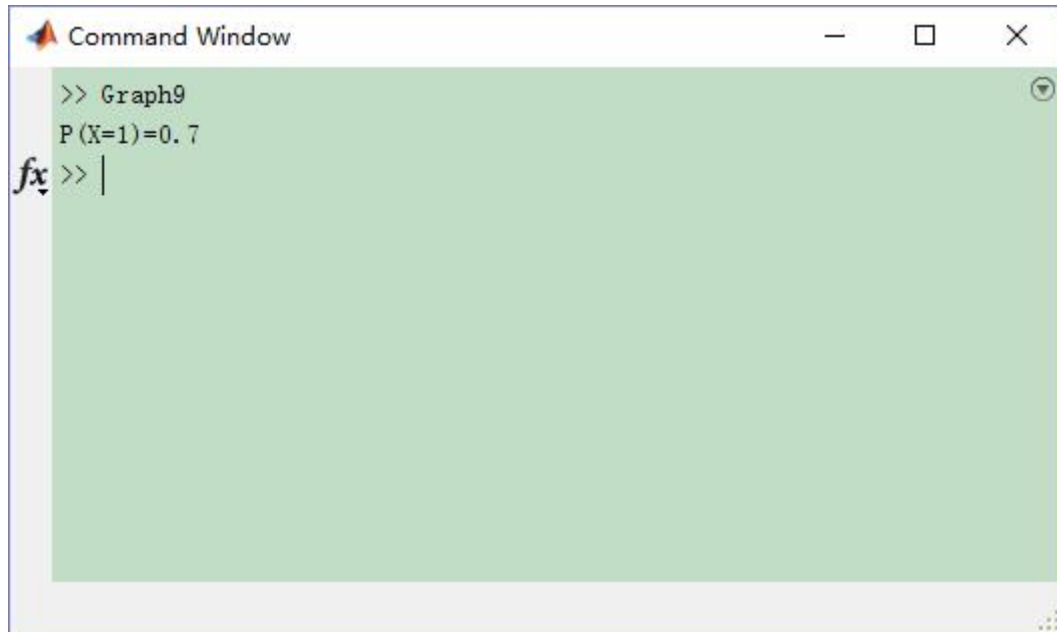
```
HX=-(r.*log2(r)+(1-r).*log2(1-r));
```

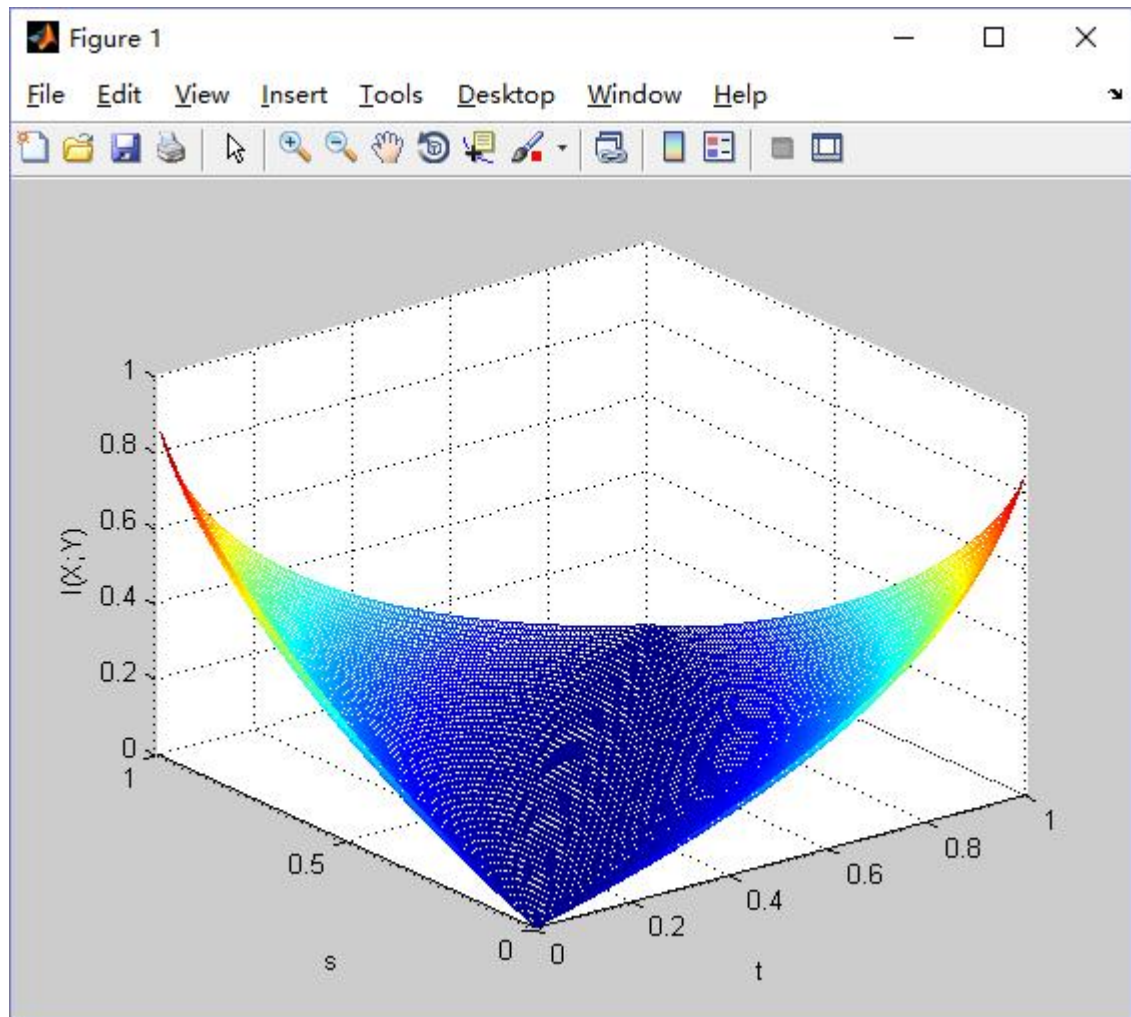
```
HX_Y=(-T.*r.*log2(T.*r./(T.*r+S.*(1-r)))...
+(-(1-T).*r.*log2((1-T).*r./((1-T).*r+(1-S).*(1-r))))...
+(-S.*(1-r).*log2(S.*(1-r)./(T.*r+S.*(1-r))))...
```



```
+(-(1-S).* (1-r). *log2((1-S)*(1-r). / ((1-T). *r+(1-S). * (1-r)))));  
mesh(T,S,HX-HX_Y);
```

运行结果:





五、实验体会

(请认真填写自己的真实体会)

1. JPanel 是 Java 图形用户界面(GUI)工具包 swing 中的面板容器类, 包含在 javax.swing 包中, 是一种轻量级容器, 可以加入到 JFrame 窗体中。JPanel 默认的布局管理器是 FlowLayout, 其自身可以嵌套组合, 在不同子容器中可包含其他组件(component), 如 JButton、JTextArea、JTextField 等, 功能是对窗体上的这些控件进行组合。

2. Java 中 JFrame 和 JPanel 的区别: JFrame 是一个窗口, 是基于系统产生的, 默认情况下, 右上角会存在最小化、最大化、关闭按钮和一个 Frame 边框; 而 JPanel 是一个面板, 是 Swing 提供的一个可以用来画图、放置其他控件的轻量级容器。

3. 对于函数的绘制, MATLAB 功能强大, 操作方便, Java 图形绘制优势在于其能够按需要添加多种组件, 灵活性较大。

4. 本实验尝试使用 Java 平台绘图, 对于函数(1)-(7)可以在 Java 中方便汇出, 但是对于函数(8)(9)使用 MATLAB 更为方便, 尤其是(9)为多维图像, Java 中绘制并不是明智的选择。

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)