模式识别上机实验 3: 密度的非参数估计

专业： 信息与计算科学 学号： 20131910023 姓名： 金洋

理论：已知一维样本 $x_1,...,x_N$，给出非参数密度估计的 Parson 窗算法。

实践：分别产生 1、16、256 和 16384 个服从一维标准正态分布的样本，

1．分别就窗宽为 $h_1 = 0.25,\ 1,\ \ 4$，$h_N = h_1/\sqrt{N}$，窗函数为高斯函数的情形估计所给样本的密度函数并画出图形。

2．分别就 $k_N = \sqrt{N}$ 时用 k_N 近邻方法估计所给样本的密度函数并画出图形。

一、算法介绍：

当 $\mu = 0$，$\sigma = 1$ 时的正态分布是标准正态分布，即

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

因此 N 个变量的一维标准正态分布的样本 x 可通过以下命令产生

x=normrnd(0,1,1,N);

为了估计 **x** 点的密度，我们可以构造一串包括 **x** 的区域 R_1，R_2，…，R_N，对 R_i 采用 i 个样本进行估计。设 V_i 是 R_i 的体积，k_i 是落在 R_i 中的样本数，$\hat{p}_i(\mathbf{x})$ 是 $p_i(\mathbf{x})$ 的第 i 次估计，则

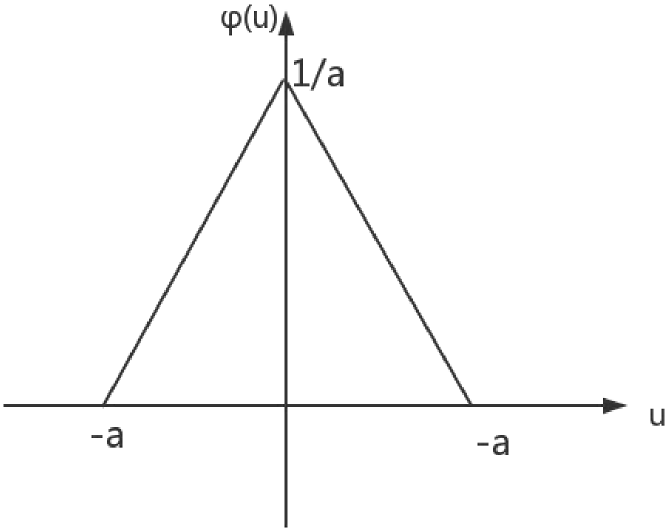
$$\hat{p}_i(\mathbf{x}) = \frac{k_i}{NV_i} \tag{1}$$

同时 V_i 和 k_i 要满足一定条件。满足这些条件的区域有两种方法：

（一） Parzen 窗法

首先引入窗函数（一维）一般有以下几种：

①三角窗函数

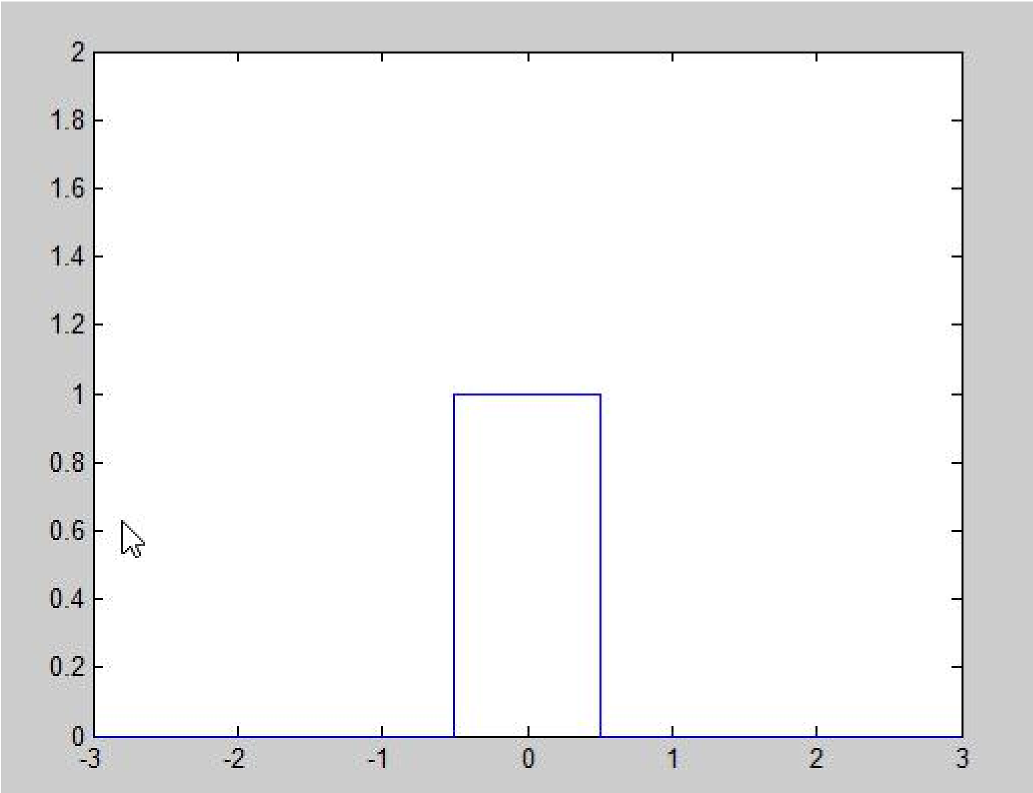


该窗函数非 0 函数值部分可由函数 y=|u|经拉伸、旋转、平移变换而来，所以窗函数表达式为

$$\varphi(u)=\begin{cases}-\dfrac{1}{a^2}|u|+\dfrac{1}{a}, & |u|\le a\\ 0, & otherwise.\end{cases}$$

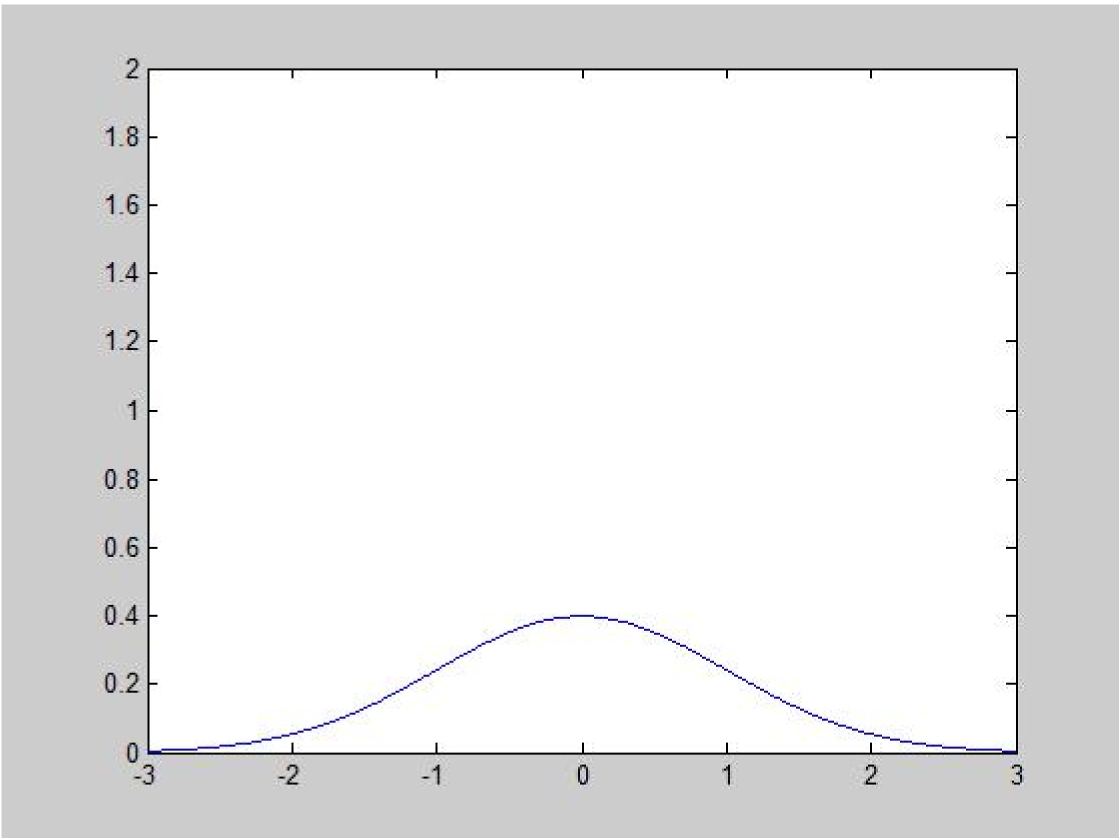
②方窗函数

$$\varphi(u)=\begin{cases}1, & |u|\le \dfrac{1}{2}\\ 0, & otherwise.\end{cases}$$



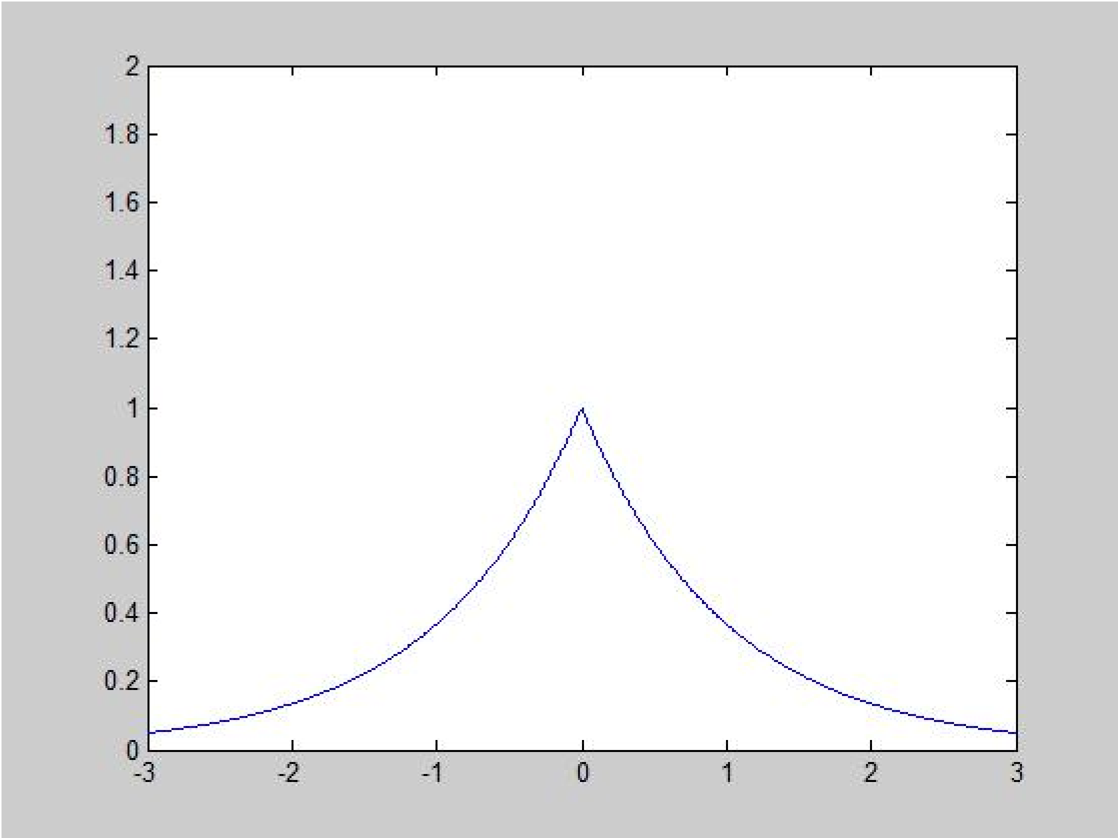
③正态窗函数

$$\varphi(u)=\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2}$$



④指数窗函数

$$\varphi(u)=e^{-|u|}$$



假设区域 R_N 是一个 d 维超立方体，h_N 为超立方体的棱长则该超立方体的体积为：

$$V_N=h_N^d$$

当 $\mathbf{x}_i$ 落入以 $\mathbf{x}$ 为中心的，体积为 V_N 的超立方体内时，$\varphi(u)=\varphi\left[\frac{\mathbf{x}-\mathbf{x}_i}{h_N}\right]=1$，否则为 0；

因此落入该超立方体内的样本数为 $k_N=\sum_{i=1}^{N}\varphi(\frac{x-x_i}{h_N})$,代入式(1),得

$$\hat{p}_N(\mathbf{x})=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{V_N}\varphi(\frac{\mathbf{x}-\mathbf{x}_i}{h_N})$$

（二） k_N -近邻估计法

k_N -近邻估计法使体积为数据的函数，而不是样本 N 的函数。例如为了从 N 个样本中估计 $p(\mathbf{x})$，我们可以预先确定 N 的某个函数 k_N，然后在 x 点的周围选择一个体积，

并让它不断增长直至捕获k_N个样本为止，这些样本为 x 的k_N个近邻。

例如取$k_N = k_1\sqrt{N}$，则式(1)变化为

$$\hat{p}_N(\mathbf{x}) = \frac{k_1}{\sqrt{N}V_N(\mathbf{x})}$$

二、实验过程：

1、程序源代码

① Main.m

```
clc;clear;
N=16384;
x=normrnd(0,1,1,N);%生成一维样本数据
y=zeros(1,N);

t=-4:0.05:4;%自变量

in=input('请输入题目编号');
if (in==1)%第一题
    h1=[0.25 1 4];%h1 分别取不同的
    %根据不同的 h1 画三个子图
    subplot(1,3,1)
    plot(x,y,'.');
    title('h1=0.25')
    hold on
    subplot(1,3,2)
    plot(x,y,'.');
    title('h1=1')
    hold on
    subplot(1,3,3)
    plot(x,y,'.');
    title('h1=4')
    hold on

    for k=1:3%每个子图画出图形
        for i=1:length(t)
            Px(i)=Parzen(t(i),x,h1(k),N);
        end
        subplot(1,3,k);
        plot(t,Px);
    end
else%第二题
```

```
    plot(x,y,'.');hold on
    for i=1:length(t)
        Px(i)=KNnn(t(i),x,N);
    end
    plot(t,Px);
end

hold off
```

② Fai.m

```
function y = Fai( u ) %通过注释可以选择不同的窗函数

    %y=abs(u)<=0.5;%方窗函数
    y=1/sqrt(2*pi)*exp(-1/2*u*u); %正态窗函数
    %y=exp(-abs(u));%指数窗函数
end
```

③ Parzen.m

```
function px=Parzen(t,x,h1,N)
    sum=0;
    hn=h1/sqrt(N);
    for i=1:N
        sum=sum+Fai((t-x(i))/hn);
    end
    px=sum/N/hn;
end
```

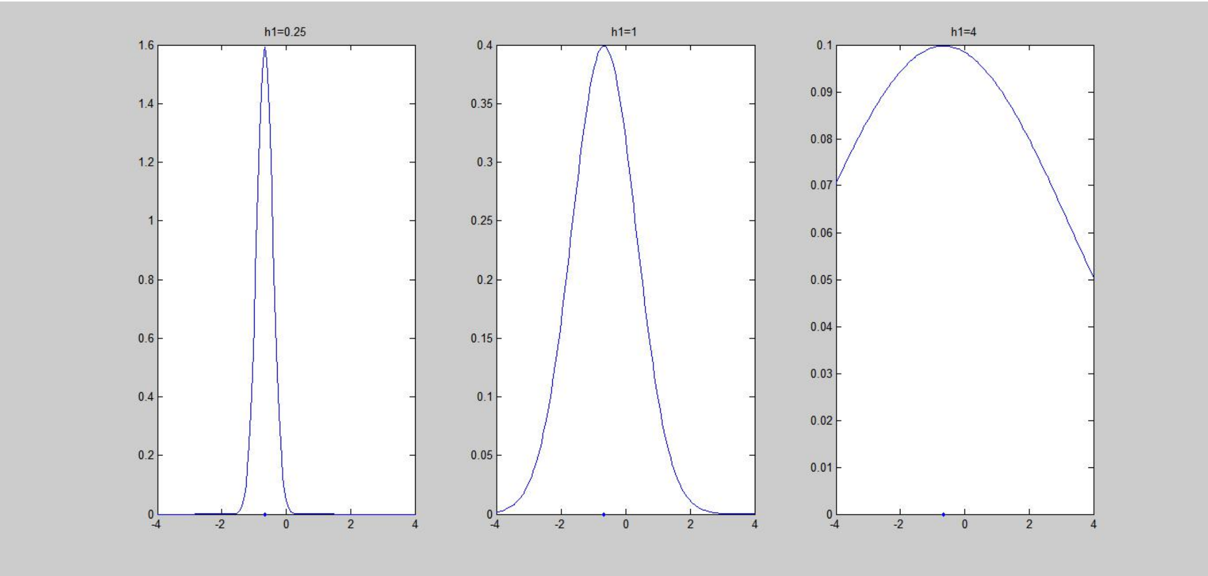
④ KNnn.m

```
function px = KNnn(t,x,N )
    kN=floor(sqrt(N));
    sortDis=sort(abs(x-t));%把样本 离自变量 t 的距离 进行排序
    VN=sortDis(kN)*2; %扩大体积，当体积包含 kN 个样本停止，由于是一维，只需*2 即可
    px=kN/N/VN;
end
```

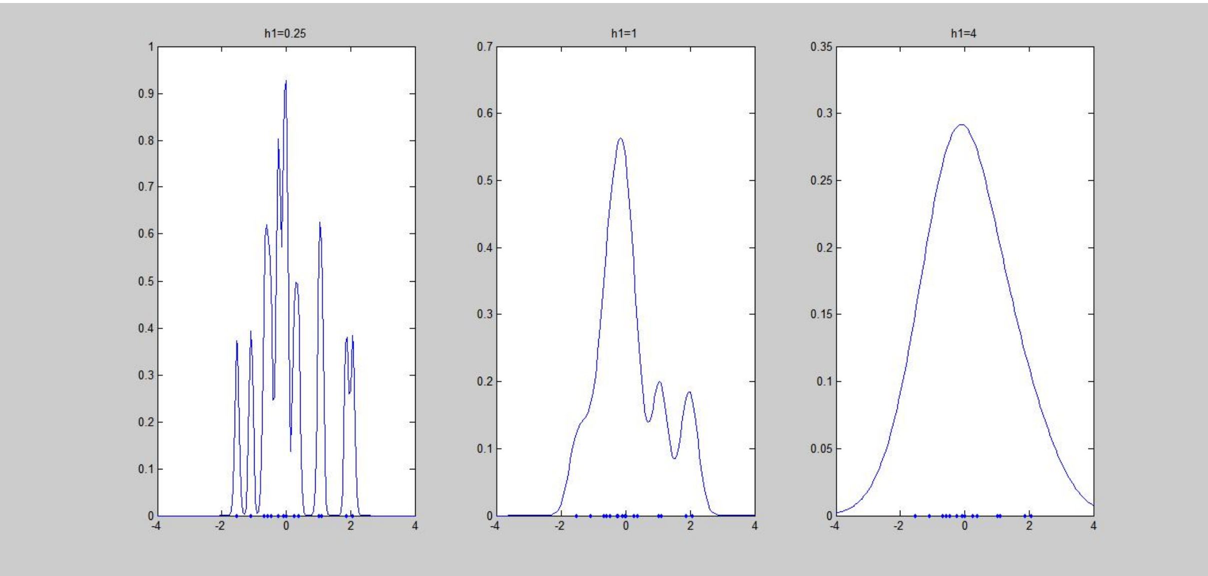
2、运行结果

（1）分别就窗宽为 $h_1 = 0.25, 1, \ 4$，$h_N = h_1 / \sqrt{N}$，窗函数为高斯函数的情形估计所给样本的密度函数并画出图形。

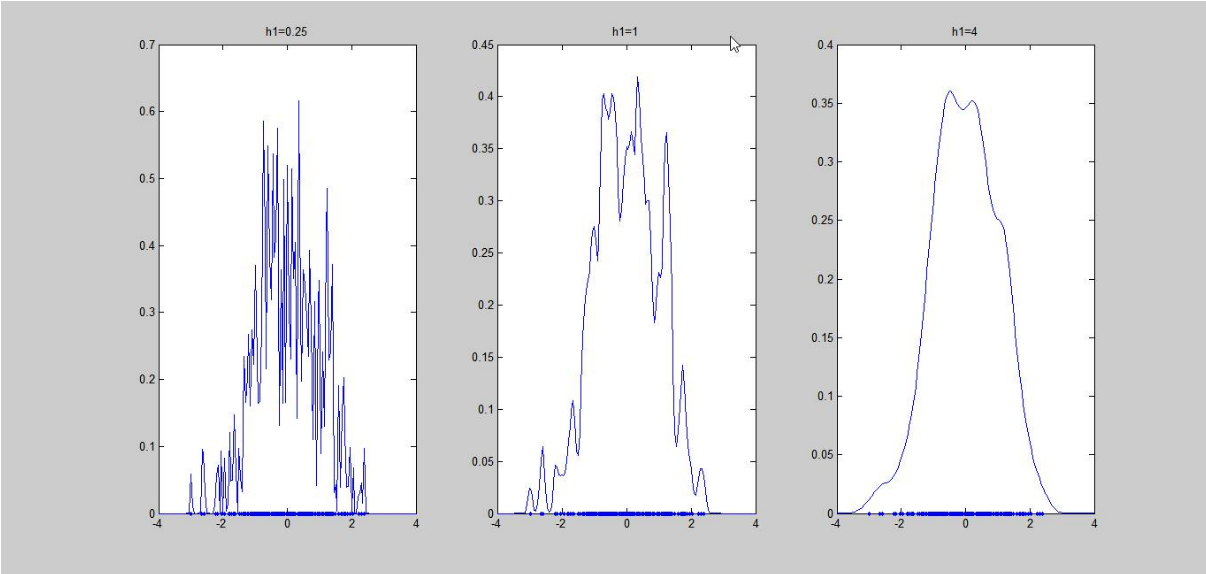
①当 N=1 时，



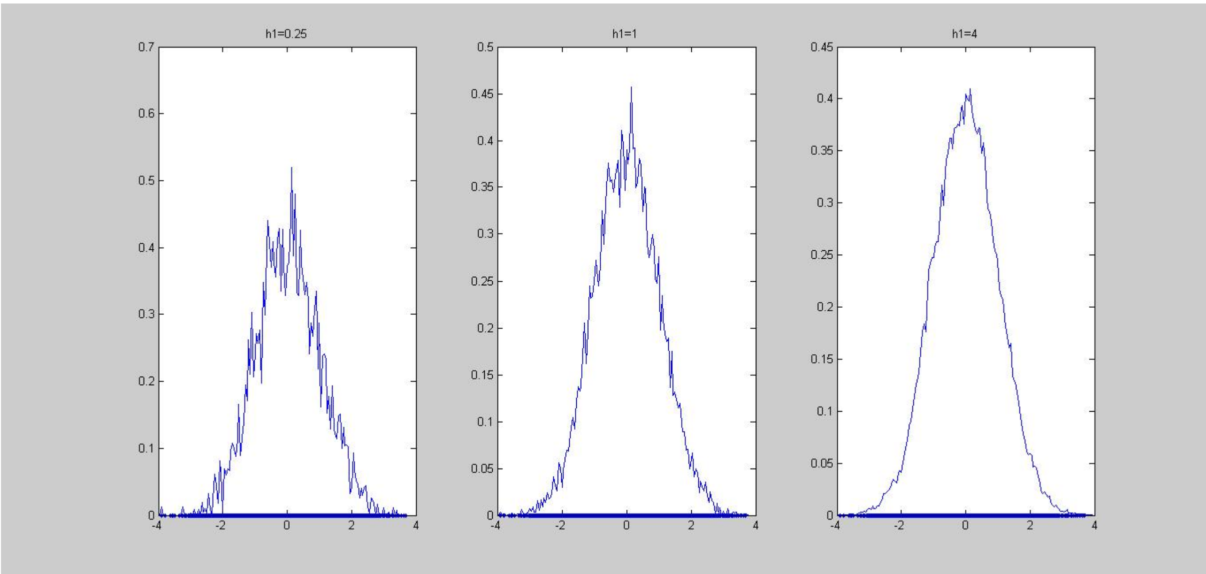
②当 N=16 时



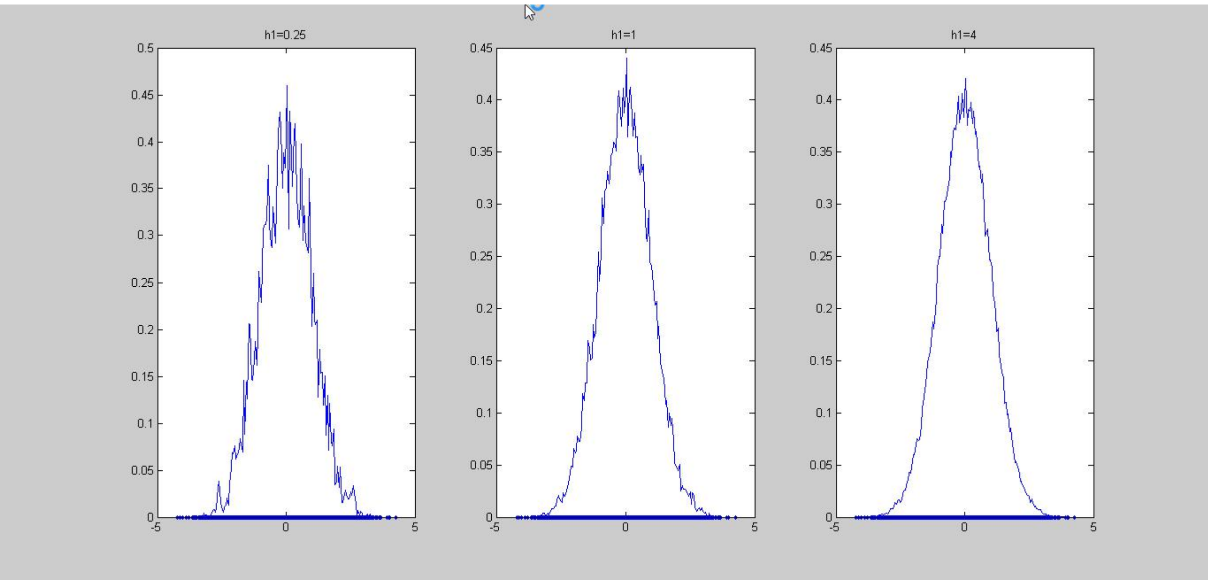
③N=256



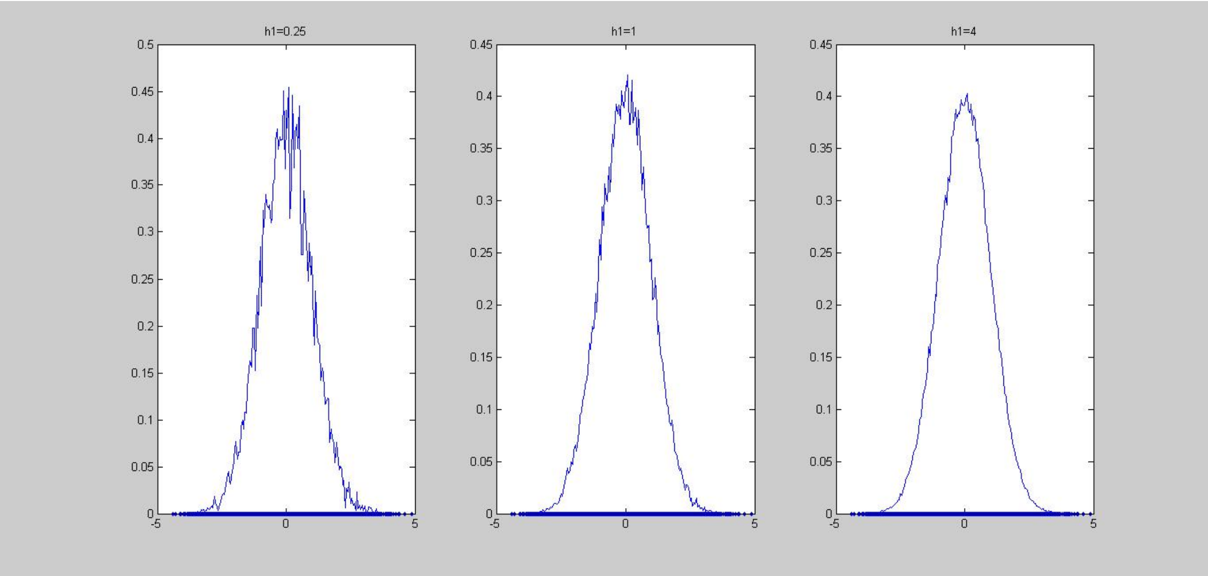
④ N=16384



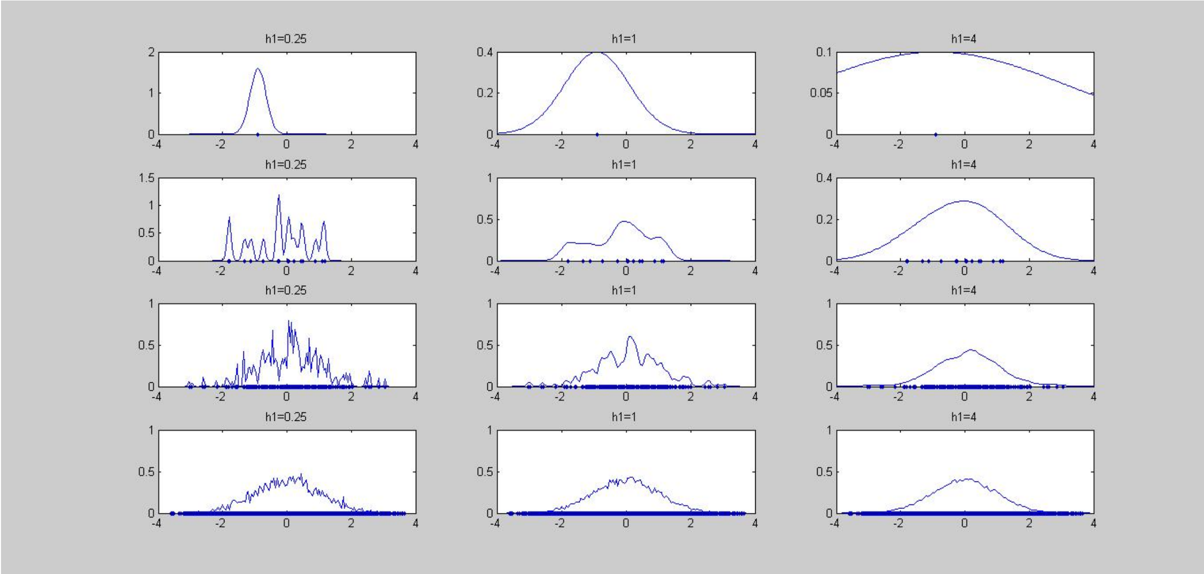
⑤ N=65536



⑥在 N=2^18 时，出结果已经很慢

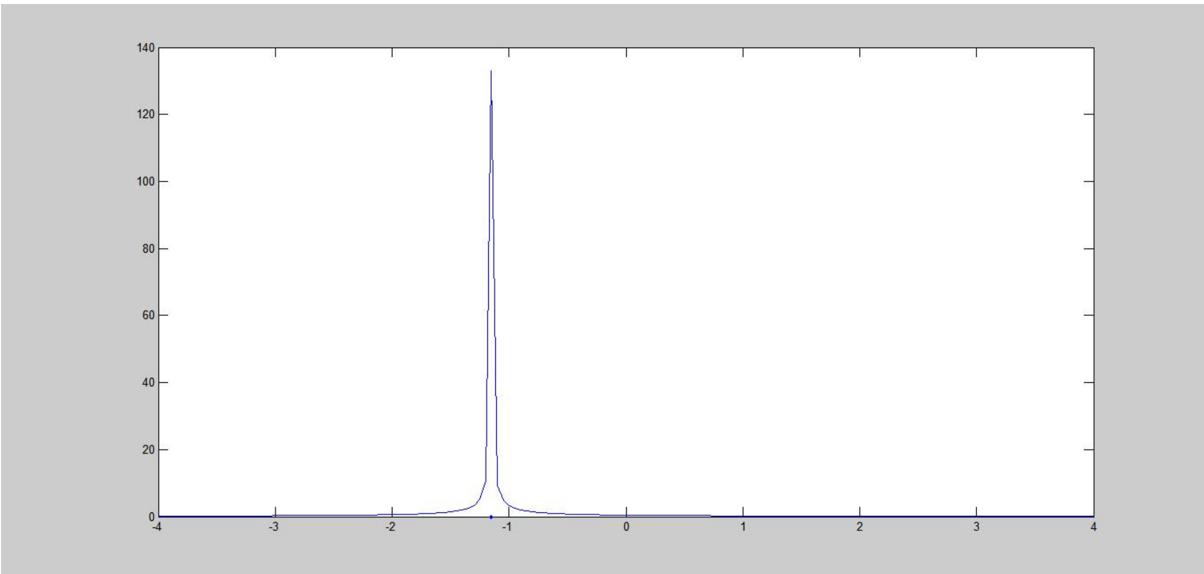


汇总后如下 ：

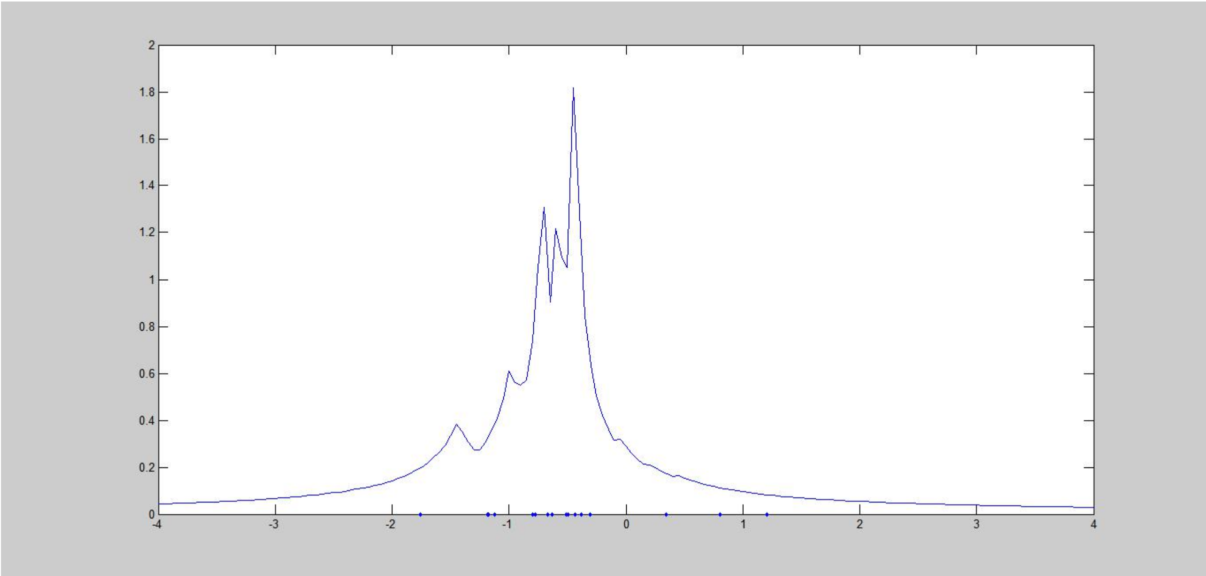


（2）分别就$k_N = \sqrt{N}$时用k_N近邻方法估计所给样本的密度函数并画出图形。

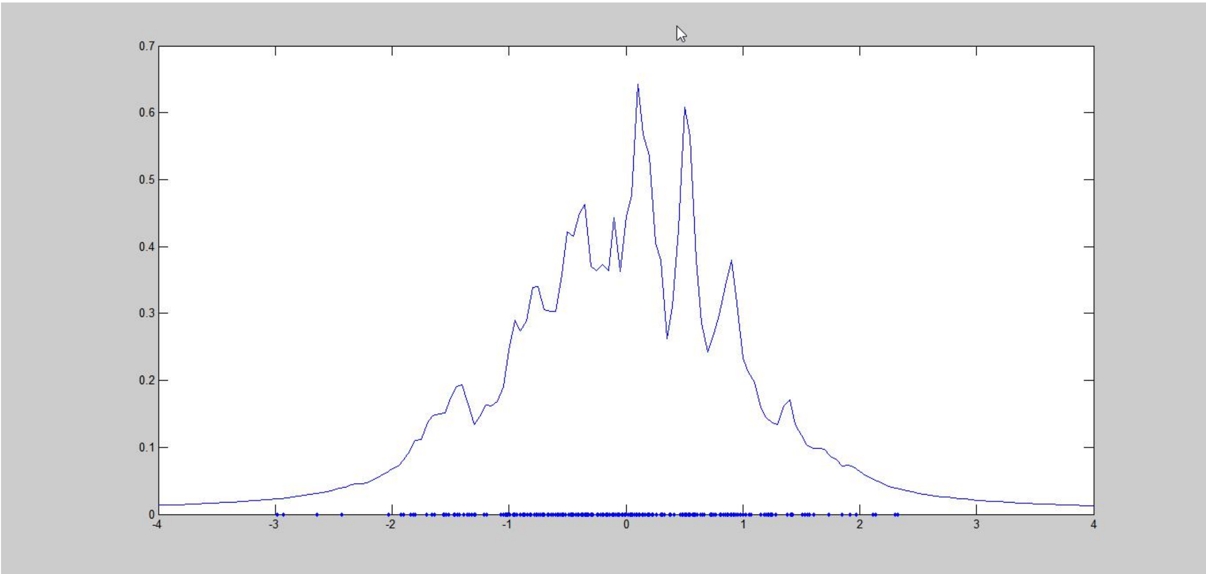
①当 N=1 时，



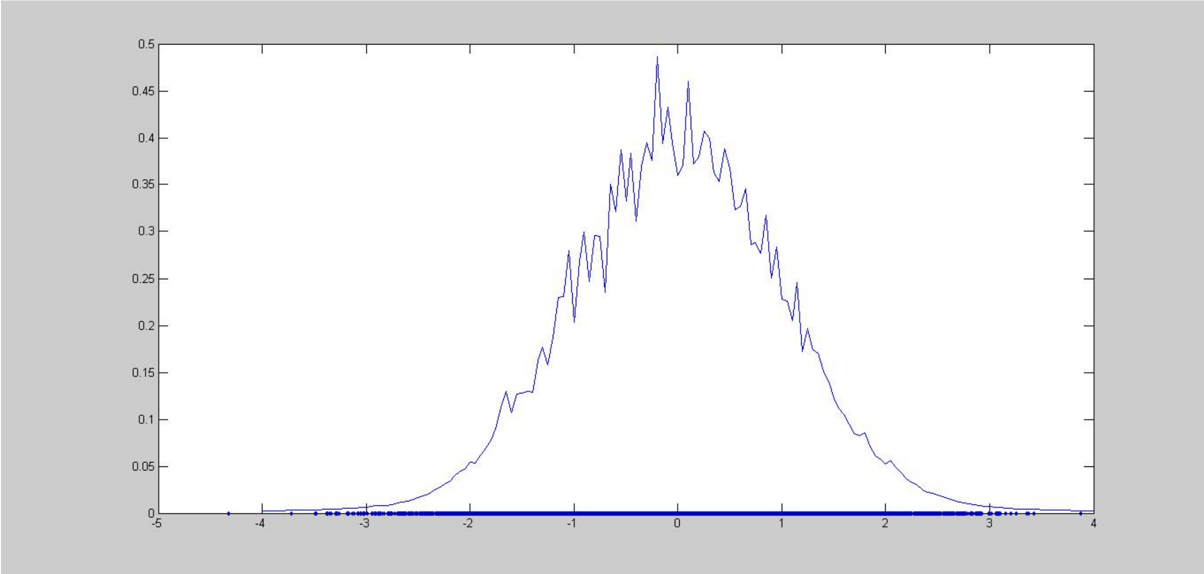
②当 N=16 时



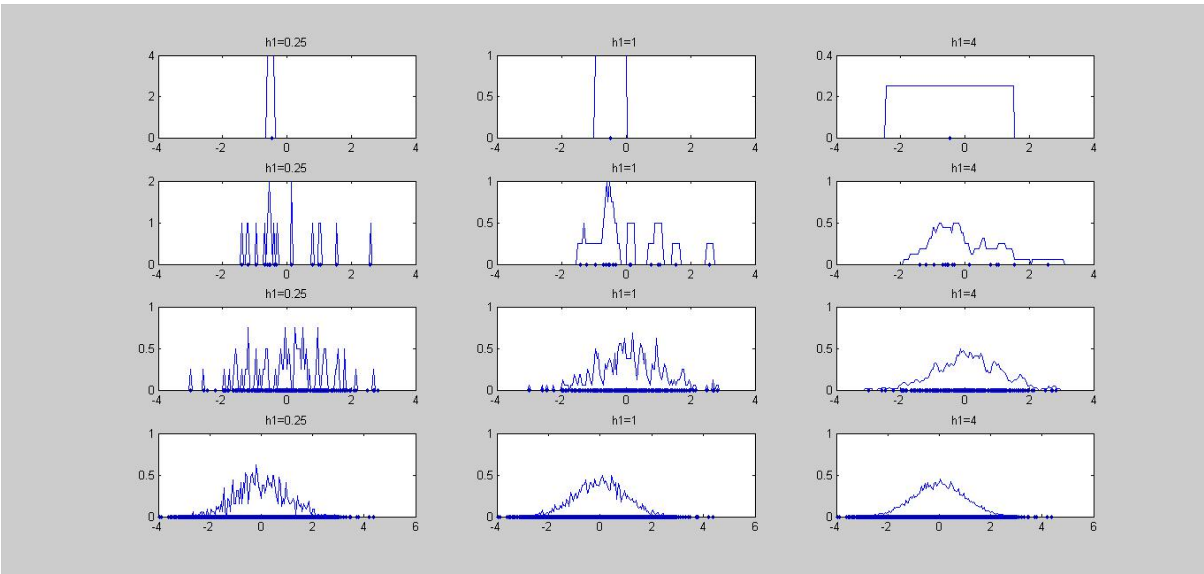
③N=256



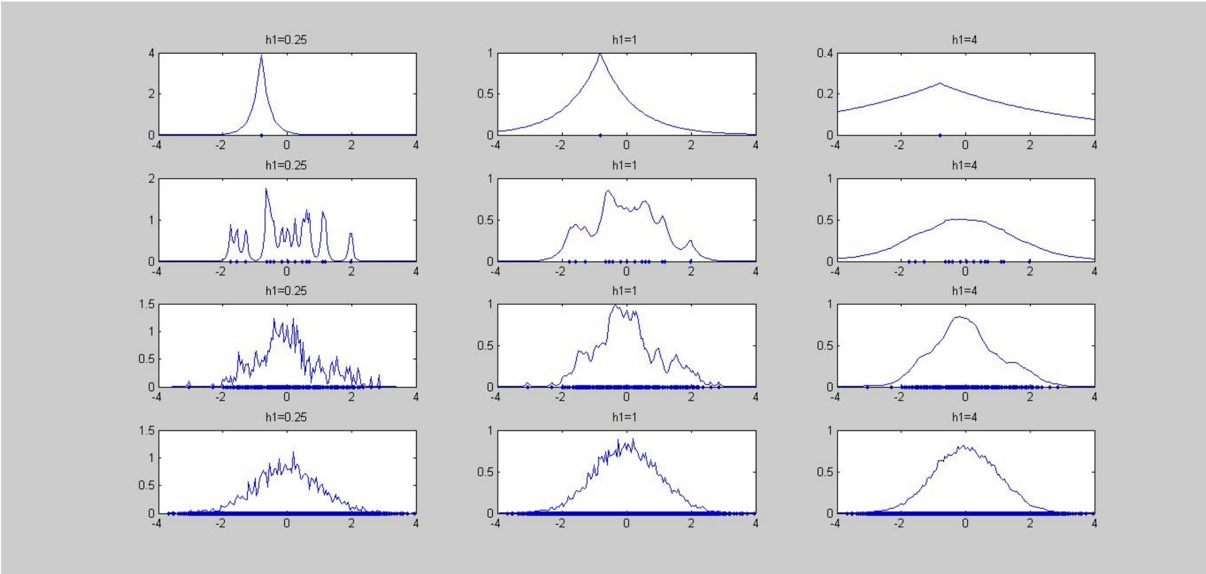
④ N=16384



（3）对（1）小题，若窗函数选为方窗函数，N 仍旧为 1、16、256、16384，将图像画在一起，则为如下：



（4）对（1）小题，若窗函数选为指数窗函数，N 仍旧为 1、16、256、16384，将图像画在一起，则为如下：


h1=0.25
h1=1
h1=4
h1=0.25
h1=1
h1=4
h1=0.25
h1=1
h1=4
h1=0.25
h1=1
h1=4