Statistical Computing

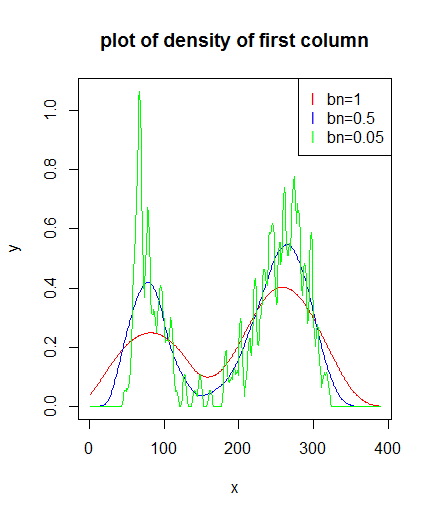
610611104 Pan,Hsing-Cheng

HW5-2

首先呼叫出faithful資料然後觀察一下資料，然後kernel選用Eparechnilcov kernel 來估計f\_hat。

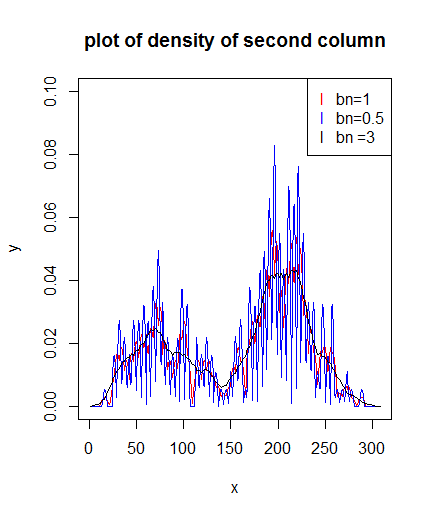
首先是first column:

|  |  |  |
| --- | --- | --- |
| Bn=1 | N=272 | X=(max-min)/272 (389) |
| Bn=0.5 | N=272 | X=(max-min)/272 |
| Bn=0.05 | N=272 | X=(max-min)/272 |



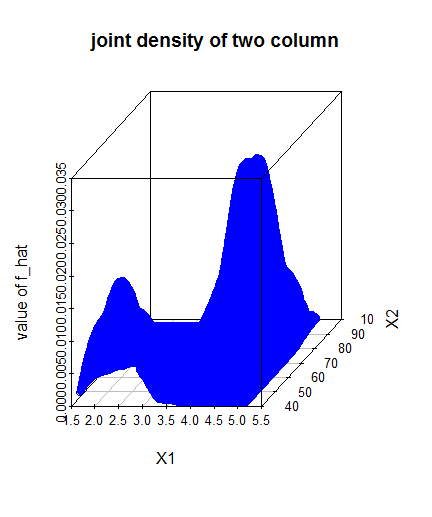
Second column:

|  |  |  |
| --- | --- | --- |
| Bn=1 | N=272 | X=(max-min)/272 |
| Bn=0.5 | N=272 | X=(max-min)/272 |
| Bn=3 | N=272 | X=(max-min)/272 |



利用最後給的P dimension kernel function 估計joint density。

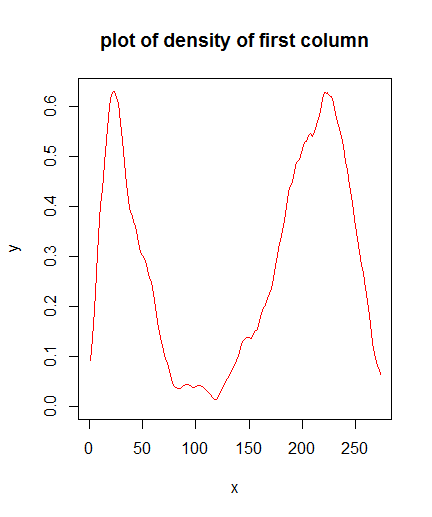
樣本點是273\*273。這邊我的B1直接取1，B2直接取3。



接續HW5-2利用persudo likelihood method求出bn。

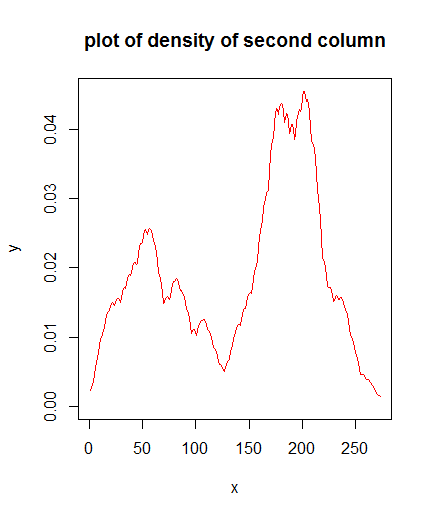
First column我是利用慢慢待得方式去收集資料來比較大小，然後我的bn從0.01開始每隔0.01帶入一次直到1。

求得當bn=0.21有最大值1.161123e-118。

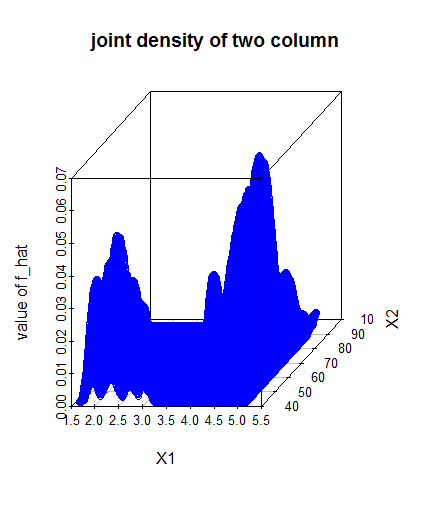


Second column因為直接帶persudo公式會發生其中一個f\_hat是0導致相乘後整個公式都是0，所以利用取log把相乘變成相加，這樣就解決了出現0之後導致相乘之後成0的問題。一樣bn從0.01開始每隔0.01帶一次直到3。

求得當bn=2.48 有最大值-1038.095。



經過上面對個別做persudo之後，嘗試利用取得的bn來畫joint kernel的圖。



2-1

|  |
| --- |
| rm(list=ls())  graphics.off()  data<-faithful  data1<-data[,1]  data2<-data[,2]  ker<-function(x,xi,n,bn)  {  to<-0  for(i in xi){  v <- abs((x-i)/bn)  if(v < 1){  s<-(3/4)\*(1-v^2)  to<-to+s  }  }  return(1/(n\*bn)\*to)  }  collect<-function(bn){  j1<-seq(1,6,(max(data1)-min(data1))/272)  c1<-c()  for(i in j1)  {  a<-ker(i,data1,length(data1),bn)  c1<-c(c1,a)  }  return(c1)  }  f1<-collect(1)  f2<-collect(0.5)  f3<-collect(0.05)  plot(f1,col="red",type = "l",ylim =c(0,max(f3)),main ="plot of density of first column",ylab = "y",xlab = "x")  legend("topright", # 表示在右上角  pch = "l", # pch代表點的圖案  col = c("red", "blue", "green"), # col代表顏色  legend = c("bn=1", "bn=0.5", "bn=0.05") # 顏色所對應的名稱  )  points(f2,type ="l",col ="blue")  points(f3,type ="l",col ="green")  ####  collect2<-function(bn){  j2<-seq(40,100,(max(data2)-min(data2))/272)  c2<-c()  for(i in j2)  {  a<-ker(i,data2,length(data2),bn)  c2<-c(c2,a)  }  return(c2)  }  g1<-collect2(1)  g2<-collect2(0.5)  g3<-collect2(0.05)  g4<-collect2(3)  plot(g1,col="red",type = "l",ylim =c(0,0.1),main ="plot of density of second column",ylab = "y",xlab = "x")  legend("topright", # 表示在右上角  pch = "l", # pch代表點的圖案  col = c("red", "blue","black"), # col代表顏色  legend = c("bn=1", "bn=0.5", "bn =3") # 顏色所對應的名稱  )  points(g2,type ="l",col ="blue")  # points(g3,type ="l",col ="green")  points(g4,type = "l",col ="black")  ### |

2-1 persudo

|  |
| --- |
| rm(list=ls())  graphics.off()  data<-faithful  data1<-data[,1]  data2<-data[,2]  ker<-function(x,xi,n,bn)  {  to<-0  for(i in xi){  v <- abs((x-i)/bn)  if(v < 1){  s<-(3/4)\*(1-v^2)  to<-to+s  } else{to<-to}  }  return(1/(n\*bn)\*to)  }  j1<-data1  b1<-seq(0.01,1,by =0.01)  c1<-c()  for(bn in b1){  s<-1  for(i in 1:length(j1))  {  a<-ker(j1[i],data1[-i],length(data1),bn)  s<-s\*a  }  c1<-c(c1,s)  }  cat("the situation of max of persudo likelihood:",which(c1==max(c1)),"max value is:",max(c1),"\n")  cat("from above to find bn:",b1[which(c1==max(c1))],"\n")  collect<-function(bn){  j1<-seq(min(data1),max(data1),(max(data1)-min(data1))/272)  c1<-c()  for(i in j1)  {  a<-ker(i,data1,length(data1),bn)  c1<-c(c1,a)  }  return(c1)  }  f1<-collect(b1[which(c1==max(c1))])  plot(f1,col="red",type = "l",ylim =c(0,max(f1)),main ="plot of density of first column",ylab = "y",xlab = "x")  #######  rm(list=ls())  graphics.off()  data<-faithful  data1<-data[,1]  data2<-data[,2]  ker<-function(x,xi,n,bn)  {  to<-0  for(i in xi){  v <- abs((x-i)/bn)  if(v < 1){  s<-(3/4)\*(1-v^2)  to<-to+s  } else{to<-to}  }  return(1/(n\*bn)\*to)  }  j2<-data2  b2<-seq(0.01,3,by =0.01)  c2<-c()  for(bn in b2){  s<-0  for(i in 1:length(j2))  {  a<-ker(j2[i],data2[-i],length(data2),bn)  s<-s+log(a)  }  c2<-c(c2,s)  }  cat("the situation of max of persudo likelihood:",which(c2==max(c2)),"max value is:",max(c2),"\n")  cat("from above to find bn:",b2[which(c2==max(c2))],"\n")  collect2<-function(bn){  j2<-seq(min(data2),max(data2),(max(data2)-min(data2))/272)  c1<-c()  for(i in j2)  {  a<-ker(i,data2,length(data2),bn)  c1<-c(c1,a)  }  return(c1)  }  g1<-collect2(b2[which(c2==max(c2))])  plot(g1,col="red",type = "l",ylim =c(0,max(g1)),main ="plot of density of second column",ylab = "y",xlab = "x") |

2-2

|  |
| --- |
| rm(list=ls())  graphics.off()  data<-faithful  data1<-data[,1]  data2<-data[,2]  ker2<-function(x)  {  if(abs(x)<1)  { return((3/4)\*(1-x^2))} else{return(0)}  }  data<-as.matrix(data)  n<-dim(data)[1]  j31<-seq(min(data1),max(data1),(max(data1)-min(data1))/272)  j32<-seq(min(data2),max(data2),(max(data2)-min(data2))/272)  bn1<-1 ;bn2<-3  # m<-matrix(0,nrow = length(j31),ncol = 273)  c3<-c()  for (l in j31) {  for(k in j32)  {  to<-0  for(n in 1:272)  {  a<-(ker2((l-data[n,1])/bn1)/bn1)\*(ker2((k-data[n,2])/bn2)/bn2)  to<-to+a  }  c3<-c(c3,to/n)  }  }  d<-c()  for(i in 1:length(j31))  {  s<-cbind(j31,j32[i])  d<-rbind(d,s)  }  library(scatterplot3d)  scatterplot3d(d[,1],d[,2],c3 ,color = "blue",type = "p",xlab = "X1",ylab = "X2",zlab = "value of f\_hat",main = "joint density of two column") |

5-1

|  |
| --- |
| rm(list=ls())  graphics.off()  f<-function(n)  {  pbn<-((3/5)/(n\*1/25\*0.211571))^(1/5)  MISE<-(pbn)^4/100\*0.211571 + 1/(n\*pbn)\*3/5  return(MISE)  }  c<-numeric(100)  for(i in 1:100)  {  c[i]<-f(50\*i)  }  op<-function(x)  {  x^-1  }  plot(op,xlim = c(0,4500),ylim = c(0,0.02))  for(i in 1:100)  {  points(x=i\*50,y=c[i])  } |