STAT 824 hw 02
Multivariate Taylor expansion, closeness of points in high-dimensional space, Nadaraya-Watson and local polynomial estimators, CV for bandwidth selection

1. For a function $f : \mathbb{R}^d \to \mathbb{R}$, let

$$
\nabla f(x_0) = \left.\begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \vdots \\ \frac{\partial}{\partial x_d} f(x) \end{bmatrix}\right|_{x=x_0}
\quad \text{and} \quad
\nabla^2 f(x_0) = \left.\begin{bmatrix} \frac{\partial^2}{\partial x_1^2} f(x) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_d} f(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_d \partial x_1} f(x) & \cdots & \frac{\partial^2}{\partial x_d} f(x) \end{bmatrix}\right|_{x=x_0} .
$$

For $x, x_0 \in \mathbb{R}^d$, show that

$$
\sum_{|\alpha| \leq 2} \frac{D^\alpha f(x_0)}{\alpha!} (x - x_0)^\alpha = f(x_0) + [\nabla f(x_0)]^T (x - x_0) + \frac{1}{2}(x - x_0)^T [\nabla^2 f(x_0)](x - x_0),
$$

which is the second-order Taylor expansion of f around x_0 evaluated at x.

> **Solution:** Interpreting the multi-index notation carefully gives
>
> $$
> \sum_{|\alpha|=0} \frac{D^\alpha f(x_0)}{|\alpha|}(x - x_0)^\alpha = f(x_0)
> $$
>
> $$
> \sum_{|\alpha|=1} \frac{D^\alpha f(x_0)}{|\alpha|}(x - x_0)^\alpha = \sum_{j=1}^{d} \left[\left.\frac{\partial}{\partial x_j} f(x)\right|_{x=x_0}\right](x_j - x_{0j})
> $$
>
> $$
> = [\nabla f(x_0)]^T (x - x_0)
> $$
>
> $$
> \sum_{|\alpha|=2} \frac{D^\alpha f(x_0)}{|\alpha|}(x - x_0)^\alpha = \frac{1}{2}\sum_{j=1}^{d}\sum_{k=1}^{d} \left[\left.\frac{\partial^2}{\partial x_j x_j} f(x)\right|_{x=x_0}\right](x_j - x_{0j})(x_k - x_{0k})
> $$
>
> $$
> = \frac{1}{2}(x - x_0)^2 [\nabla^2 f(x_0)](x - x_0)
> $$

2. Obtain $n = 500$ realizations of (X_1, X_2) by running the code

```
n <- 500; alpha <- 1/3; Z <- runif(n) < alpha; X <- matrix(NA,n,2)
X[,1] <- rnorm(n,2*Z,1);X[,2] <- rnorm(n,3*Z,1)
```

Use leave-one-out crossvalidation to select the bandwidth for a bivariate kernel density estimator (write your own code for this). Then make a plot showing the CV criterion as a function of h and a scatterplot of your (X_1, X_2) values with contours of your estimate (at the CV choic of bandwidth) overlaid. Report your chosen bandwidth. My selected bandwidth was $\hat{h} = 0.4$. These are my plots:
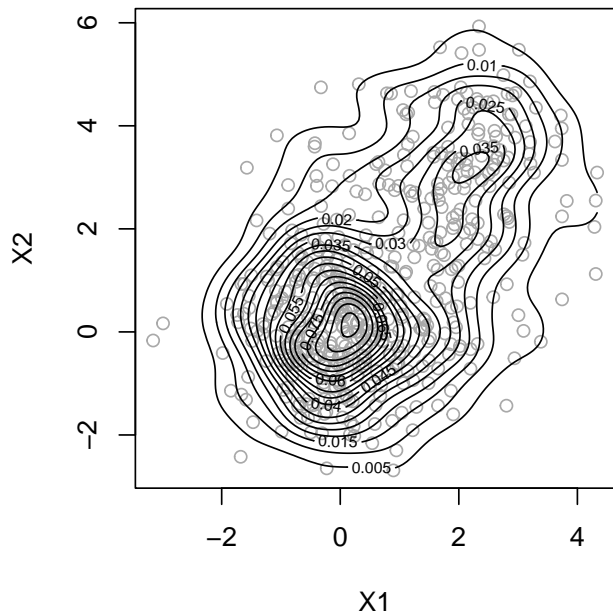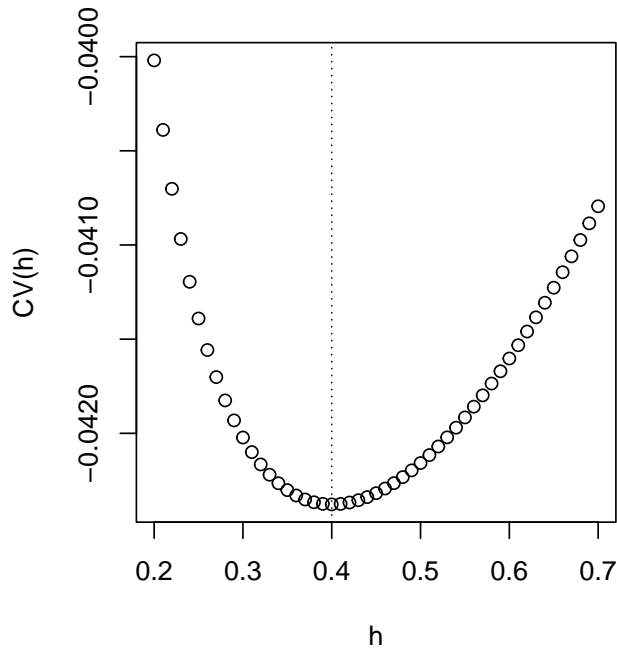
Solution: We can perform leave-one-out crossvalidation just as in the univariate case; the only complication is how we compute the integral $\int_{\mathbb{R}^2} \hat{f}_n(x)dx$. We can get a numerical approximation to this integral by computing the height of $\hat{f}_n(x)$ over a grid. See the below code:

```
n <- 500
alpha <- 1/3
Z <- runif(n) < alpha
X <- matrix(NA,n,2)
X[,1] <- rnorm(n,2*Z,1)
X[,2] <- rnorm(n,3*Z,1)

biv_kde <- function(x,Y,h){

  val <- mean(dnorm(Y[,1] - x[1],0,h) * dnorm(Y[,2]-x[2],0,h))
  return(val)

}


hh <- seq(.2,.7,by=.01)
gridsize <- 120
x1.seq <- seq(min(X[,1]),max(X[,1]),length = gridsize)
x2.seq <- seq(min(X[,2]),max(X[,2]),length = gridsize)
zz <- matrix(0,gridsize,gridsize)

CV <- numeric(length(hh))
for(k in 1:length(hh)){
```

```r
  h <- hh[k]

  for( i in 1:gridsize)
    for( j in 1:gridsize){

      zz[i,j] <- biv_kde(x = c(x1.seq[i],x2.seq[j]),Y = X, h = h)

    }

  Ahat <- sum(zz^2*diff(x1.seq)[1] * diff(x2.seq)[2])
  # sum(zz*diff(x1.seq)[1] * diff(x2.seq)[2]) # should be close to 1

  Bhat <- 0
  for(i in 1:n){
    fnii <- biv_kde(x = X[i,],Y = X[-i,], h = h)
    Bhat <- Bhat + 2 * fnii / n
  }

  CV[k] <- Ahat - Bhat
  print(k)

}

h_cv <- hh[which.min(CV)]

for( i in 1:gridsize)
  for( j in 1:gridsize){

    zz[i,j] <- biv_kde(x = c(x1.seq[i],x2.seq[j]),Y = X, h = h_cv)

  }

par(mfrow = c(1,2), mar = c(4.1,4.1,1.1,1.1))

plot(CV ~ hh,
     xlab = "h",
     ylab = "CV(h)")
abline(v = h_cv, lty = 3)

plot(X[,2]~X[,1],
     col = "dark gray",
     xlab = "X1",
     ylab = "X2",
     )

contour(x1.seq, x2.seq, zz, add = TRUE, nlevels = 20)
```

3. Let $X, X_1, \ldots, X_n \in [0, 1]^d$ be independent random vectors with the elements of each being independent and uniformly distributed on the interval $[0, 1]$. For a vector $x \in \mathbb{R}^d$, let $\|x\|_\infty = \max_{1 \leq k \leq d} |x_k|$.

   (a) Show that
   $$\mathbb{E} \min_{1 \leq i \leq n} \|X - X_i\|_\infty \geq \frac{d}{2(d+1)} \cdot \frac{1}{n^{1/d}}.$$

   > **Solution:** We have
   > $$P\left(\min_{1 \leq i \leq n} \|X - X_i\|_\infty \leq t \right) = P(\cup_{i=1}^n \|X - X_i\|_\infty \leq t)$$
   > $$\leq nP(\|X - X_1\|_\infty \leq t)$$
   > $$= \int_{\{\mathbf{x} \in [0,1]^d\}} \int_{\{\mathbf{x}_1 \in [0,1]^d : \|\mathbf{x} - \mathbf{x}_1\|_\infty \leq t\}} d\mathbf{x}_1 d\mathbf{x}$$
   > $$\leq \int_{\{\mathbf{x} \in [0,1]^d\}} (2t)^d d\mathbf{x}$$
   > $$= n(2t)^d,$$
   >
   > since the volume of a d-dimensional unit cube with width $2t$ in each dimension is $(2t)^d$. We then get the result by integrating over the corresponding lower bound for the survival function (where this is nonnegative). We have
   > $$\mathbb{E} \min_{1 \leq i \leq n} \|X - X_i\|_\infty \geq \int_0^{1/(2n^{1/d})} (1 - n \cdot (2t)^d) dt,$$
   >
   > which gives the bound.

   (b) Give an interpretation of the claim.

   > **Solution:** As the dimension of the space in which the data lie grows, the far-between-ness of the points grows, such that to maintain a dense "cloud" of points in a higher and higher dimensional space, one must increase the number of points *extremely* fast.

4. For a set of points $(X_1, Y_1), \ldots, (X_n, Y_n)$, the Nadaraya-Watson estimator of $m(x) = \mathbb{E}[Y|X = x]$ is
   $$\hat{m}_n^{\text{NW}}(x) = \sum_{i=1}^n W_{ni}(x) Y_i, \quad \text{with} \quad W_{ni}(x) = \frac{K(h^{-1}(X_i - x))}{\sum_{j=1}^n K(h^{-1}(X_j - x))}.$$

   (a) Show that if $K \geq 0$ we have $\hat{m}_n^{\text{NW}}(x) = \underset{\theta \in \mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^n (Y_i - \theta)^2 K(h^{-1}(X_i - x))$.

   (b) Show that
   $$\frac{Y_i - \hat{m}_n^{\text{NW}}(X_i)}{1 - W_{ni}(X_i)} = Y_i - \hat{m}_{n,-i}^{\text{NW}}(X_i).$$

Solution: We have

$$
\begin{aligned}
Y_i - \hat{m}^{\mathrm{NW}}_{n,-i}(X_i) &= Y_i - \frac{\sum_{j\neq i} Y_j K(h^{-1}(X_j - X_i))}{\sum_{k\neq i}^{n} K(h^{-1}(X_k - X_i))} \\
&= Y_i - \frac{\sum_{j=1}^{n} Y_j K(h^{-1}(X_j - X_i)) - Y_i K(h^{-1}(X_i - X_i))}{\sum_{k\neq i}^{n} K(h^{-1}(X_k - X_i))} \\
&= Y_i - \frac{\hat{m}^{\mathrm{NW}}_n(X_i) - Y_i \cdot W_{ni}(X_i)}{1 - W_{ni}(X_i)} \\
&= \frac{Y_i - \hat{m}^{\mathrm{NW}}_n(X_i)}{1 - W_{ni}(X_i)},
\end{aligned}
$$

where we obtain the third equality by dividing the numerator and denominator of the fraction by $\sum_{k=1}^{n} K(h^{-1}(X_k - X_i))$.

(c) Explain why the fact in part (b) is useful.

Solution: This is useful because it allows us to write

$$
\mathrm{CV}_n(h) = \frac{1}{n} \sum_{i=1}^{n} [Y_i - \hat{m}^{\mathrm{NW}}_{n,-i}(X_i)]^2 = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{Y_i - \hat{m}^{\mathrm{NW}}_n(X_i)}{1 - W_{ni}(X_i)} \right]^2,
$$

so that we may compute the crossvalidation prediction risk without actually doing crossvalidation computationally; this saves time.

5. For $n = 200$, generate data according to $Y_i = m(X_i) + \varepsilon_i$, $i = 1,\ldots,n$, where $X_1,\ldots,X_n \overset{\mathrm{ind}}{\sim}$ Beta$(1/2, 1/2)$, independent of $\varepsilon_1,\ldots,\varepsilon_n \overset{\mathrm{ind}}{\sim}$ Normal$(0,1)$, where

$$
m(x) = -250 \cdot (x - 1/2) \cdot \phi\left(10(x - 1/2)\right), \quad \phi(z) = (1/\sqrt{2\pi})e^{-z^2/2}.
$$

Choose via crossvalidation a value of the bandwidth h for the local linear estimator (local polynomial of order $\ell = 1$) using ϕ as the kernel function. *Note: You will have to specify a grid of candidate h values.*

(a) Make a plot of the function

$$
\mathrm{CV}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{Y_i - \hat{m}^{\mathrm{LP}}_{n,1}(X_i)}{1 - W^*_{ni}(X_i)} \right]^2
$$

over your grid of candidate bandwidths. It should dip down and rise back up. The weights $W^*_{ni}(X_i)$ are the values such that $\hat{m}^{\mathrm{LP}}_{n,1}(X_i) = \sum_{i=1}^{n} W^*_{ni}(X_i)Y_i$.

(b) Make a scatterplot of the data and overlay the true function; include in the scatterplot the estimated function at your chosen value of the bandwidth.

(c) Turn in your code.

Solution: My plot looks like