Nonparametric Regression

Badr Missaoui

Nonparametric Regression

Outline

- Kernel and local polynomial regression.
- Penalized regression.

Nonparametric Regression

- We are given n pairs of observations (X_1, Y_1),...,(X_n, Y_n) where

$$Y_i = r(X_i) + \varepsilon_i, \quad i = 1, ..., n$$

  and

$$r(x) = \mathbb{E}(Y|X = x).$$

- If X_i are deterministic, we assume that $\varepsilon_i \sim N(0, \sigma^2)$.
- If X_i are random variables, we assume that $\varepsilon_i \sim N(0, \sigma^2)$ are independent of X_i.
- In the absence of any hypothesis on the function r, we are in the nonparametric framework.

Nonparametric Regression

- The simplest nonparametric estimator is the **regressogram**.
- Suppose that X_i are in the interval $[a, b]$ and denote the bins by$B_1, ..., B_m$. Let k_j be the number of observations in bin B_j.
- Define

$$\hat{r}_n(x) = \frac{1}{k_j} \sum_{i : X_i \in B_j} Y_i = \bar{Y}_j \quad \text{for } x \in B_j$$

- We can rewrite the estimator as

$$\hat{r}_n(x) = \sum_{i=1}^{n} \ell_i(x) Y_i$$

  where $\ell_i(x) = 1/k_j$ if $X_j \in B_j$, and $\ell_i(x) = 0$ otherwise.
- In other words, the estimate $\hat{r}_n$ is a step function obtained by averaging the Y_i over each bin

Nonparametric Regression

- Recall from our discussion of model selection that

  $$R(h) = \mathrm{E}(Y - \hat{r}_n(X))^2 = \sigma^2 + \mathrm{E}(r(X) - \hat{r}_n(X))^2 = \sigma^2 + MSE$$

  .
- We can write the MSE as

  $$MSE = \int \mathrm{bias}^2(x) p(x) dx + \int \mathrm{var}(x) p(x) dx$$

  where

  $$\mathrm{bias}(x) = \mathrm{E}(\hat{r}_n(x) - r(x))$$

  and

  $$\mathrm{var}(x) = \mathrm{Variance}(\hat{r}_n(x))$$
- When the data are oversmoothed, the bias term is large and the variance is small.
- When the data are undersmoothed, the opposite is true.
- This is called the **bias-variance tradeoff**.
- Minimizing risk corresponds to balancing bias and variance.

Nonparametric Regression

- Ideally, we would like to choose h to minimize $R(h)$.
- $R(h)$ depends on the unknown function $r(x)$. We use instead the average residual sums of squares

  $$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{r}_n(X_i))^2$$

  to estimate $R(h)$.
- We will estimate the risk using the leave-one-out cross validation which defined by

  $$CV = \hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{r}_{n(-i)}(X_i))^2$$

  where $\hat{r}_{n(-i)}$ is the estimator obtained by omitting the i^{th} pair (X_i, Y_i)

Nonparametric Regression

- There is a shortcut formula for computing $\widehat{R}$ just like in linear regression.
- Let $\hat{r}_n$ be a linear smoother. Then the leave-one-out cross-validation $\widehat{R}(h)$ can be written as

$$\widehat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{r}_{n(-i)}(X_i)}{1 - L_{ii}}\right)^2$$

  where $L_{ii} = \ell_i(X_i)$
- An alternative is to use generalized cross validation

$$GCV(h) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \hat{r}_n(X_i)}{1 - n^{-1}\sum_{i=1}^{n} L_{ii}}\right)^2$$

Nonparametric Regression

Kernel regression

- **Kernel** refers to any smooth function K such that $K(x) \geq 0$ and
$$\int K(x)dx = 1, \ \int xK(x)dx = 0$$
and
$$\sigma_K^2 = \int x^2 K(x)dx > 0$$
- Some commonly used kernels :
- the boxcar kernel : $K(x) = \frac{1}{2} I_{|x| \leq 1}$
- the Gaussian kernel : $K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^{2/2}}$

Nonparametric Regression

Kernel regression

- Let $h > 0$ be a positive number (bandwidth). The **Nadaraya-Watson kernel estimator** is defined by

  $$\hat{r}_n(x) = \sum_{i=1}^{n} \ell_i(x) Y_i$$

  where

  $$\ell_i(x) = \frac{K\left(\frac{x-X_i}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{x-X_i}{h}\right)}$$

- In `R`, we suggest using the `loess` command or the `locfit` library

```
out = loess(y  x,span=.25,degree=0)
lines(x,fitted(out))


out = locfit(y  x,deg=0,alpha=c(0,h))
```

Nonparametric Regression

Kernel regression

- To do the cross-validation, create a vector bandwidths $h = (h_1, ..., h_k)$

```
h = c( ... put your values here ... )
k = length(h)
zero = rep(0,k)
H = cbind(zero,h)
out = gcvplot(yx,deg=0,alpha=H)
plot(out$df,out$values)
```

Nonparametric Regression

Kernel regression

- The choice of the kernel K is not too important.
- What does matter much is the choice of the bandwidth which controls the amount of smoothing. In general the bandwidth depends on the sample size (h_n).
- We assume that f is the density of $x_1, ..., x_n$.
- The risk of the Nadaraya-Watson kernel estimator is

$$\begin{aligned} R(h_n) &= \frac{h_n^4}{4}\left(\int x^2K(x)dx\right)^2 \int \left(r''(x) + 2r'(x)\frac{f'(x)}{f(x)}\right)^2 dx \\ &+ \frac{\sigma^2 \int K^2(x)dx}{nh_n} \int \frac{1}{f(x)} dx + o(n^{-1}h_n) + o(h_n^4) \end{aligned}$$

as $h_n \to 0$ and $nh_n \to \infty$

Nonparametric Regression

Kernel regression

- What is especially notable is the presence of the term

$$2r'(x)\frac{f'(x)}{f(x)}$$

- This means that the bias is sensitive to the position's of the X_is.
- If we differentiate R with respect to h_n and set the result to 0, we find that the optimal h_* is

$$h_* = \left(\frac{1}{n}\right)^{1/5}\left(\frac{\sigma^2\int K^2(x)dx\int\frac{1}{f(x)}dx}{\left(\int x^2K(x)dx\right)^2\int\left(r''(x)+2r'(x)\frac{f'(x)}{f(x)}\right)^2dx}\right)^{1/5}$$

Nonparametric Regression

Kernel regression

- Thus, $h_* = n^{-1/5}$. The risk R_{h_n} decreases at rate $(n^{-4/5})$
- In practice we cannot not use the formula of h_* mentioned above since it depends on the unknown function r.
- Instead, we use leave-one-out cross-validation.

Nonparametric Regression

Local Polynomials

- Kernel estimators suffer from design bias.
- These problem can be alleviated by using a **local polynomial regression.**
- The idea is to approximate a smooth regression function $r(u)$ in the target value x by the polynomial :

  $$r(u) \sim P_x(u; a)$$

  where

  $$P_x(u; a) = a_0 + a_1(u - x) + \frac{a_2}{2!}(u - x)^2) + ... + \frac{a_p}{p!}(u - x)^p.$$

- We estimate $a = (a_0, ..., a_p)^T$ by minimizing

  $$\sum_{i=1}^{n} w_i(x)(Y_i - P_x(u; a))^2$$

Nonparametric Regression

Local Polynomials

- To find $\hat{a}(x)$, it is helpful to re-express the problem in matrix notation.
- Let

$$X_x = \begin{pmatrix} 1 & x_1 - x & \cdots & \frac{(x_1 - x)^p}{p!} \\ 1 & x_2 - x & \cdots & \frac{(x_2 - x)^p}{p!} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n - x & \cdots & \frac{(x_n - x)^p}{p!} \end{pmatrix}$$

  and let $W_x = diag\{w_i(x)\}_i$
- we can write then the problem as

$$(Y - X_x a)^T W_x (Y - X_x a)$$

- Minimizing this gives the weighted least squares estimator

$$a(x) = (X_x^T W_x X_x)^{-1} X_x^T W_x Y$$

Nonparametric Regression

Local Polynomials

- The local polynomial regression estimate is

  $$\hat{r}_n(x) = \sum_{i=1}^{n} \ell_i(x) Y_i$$

  where $\ell(x)^T = (\ell_1(x), ..., \ell_n(x))$,

  $$\ell(x)^T = e_1^T (X_x^T W_x X_x)^{-1} X_x^T W_x$$

  $e_1 = (1, 0, ..., 0)^T$.

- The R code is the same except we use `deg=1` for local linear, `deg=2` for quadratic

```
    loess(y  x,deg=1,span=h)
    locfit(y  x,deg = 1,alpha=c(0,h))
```

Nonparametric Regression

Local Polynomials

- Let $Y_i = r(X_i) + \sigma(X_i)\varepsilon_i$ for $i = 1, ..., n$ and $a < X_i < b$. Assume that $X_1, ..., X_n$ are a sample from a distribution with density f and that (i) $f(x) > 0$, (ii) f, r'' and σ^2 are continuous in a neighborhood of x, and (iii) $h_n \to 0$ and $nh_n \to 0$.

  The local linear estimator has a variance

  $$\frac{\sigma^2(x)}{f(x)nh_n} \int K^2(x)dx + o(1/nh_n)$$

  and has an asymptotic bias

  $$h_n^2 \frac{1}{2} r''(x) \int x^2 K(x)dx + o(h^2)$$

- Thus, the local linear estimator is free from design bias.

Nonparametric Regression

Penalized Regression

- Consider polynomial regression

$$Y = \sum_{i=0}^{p} \beta_j x^j + \varepsilon$$

  or

$$\hat{r}(x) = \sum_{j=0}^{n} \hat{\beta}_j x^j$$

- We have the design matrix

$$X = \begin{pmatrix} 1 & x_1 & \cdots & x_1^p \\ 1 & x_2 & \cdots & x_2^p \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^p \end{pmatrix}$$

Nonparametric Regression

Penalized Regression

- Least squares minimizes

$$(Y - X\beta)^T(Y - X\beta),$$

  which implies $\hat{\beta} = (X^T X)^{-1} X^T Y$
- The ridge regression aims to minimize

$$(Y - X\beta)^T(Y - X\beta) + \lambda\beta^T\beta$$

  then $\tilde{\beta} = (X^T X + \lambda I)^{-1} X^T Y$

Nonparametric Regression

Penalized Regression

- An alternative way is to minimize the penalized sums of squares
$$M(\lambda) = \sum_i (Y_i - \hat{r}_n(X_i))^2 + \lambda J(r)$$
where
$$J(r) = \int (r''(x))^2 dx$$
is the roughness penalty.
- This penalty leads to a solution that favors smoother functions.
- The parameter λ controls the amount of smoothness.

Nonparametric Regression

Penalized Regression

- The most commonly used splines are piecewise cubic splines.
- Let $\xi_1 < ... < \xi_k$ be a set of ordered point -called **knots**- contained in some interval (a, b). A **cubic spline** is a continuous function r such that (i) r is a cubic polynomial over $(\xi_1, \xi_2), ...$ and r has continuous first and second derivatives at the knots.
- A spline that is linear beyond the boundary knots is called a **natural spline**.

Theorem

*The function $\hat{r}_n(x)$ that minimizes $M(\lambda)$ is a natural cubic spline with knots the data points. The estimator $\hat{r}_n$ is called a **smoothing spline**.*

Nonparametric Regression

Penalized Regression

- The theorem above does not give you an explicit form for $\hat{r}_n$
- We will construct a basis for for the set of splines (cubic B-spline).

$$
\begin{aligned}
B_{i,0}(t) &= 1_{t \in [t_i, t_{i+1}]} \\
B_{i,d}(t) &= \frac{t - t_i}{t_{i+d} - t_i} B_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(t)
\end{aligned}
$$

- Without the penalty, the B-spline basis interpolate the data and therefore provide a perfect fit to the data.

Nonparametric Regression

Penalized Regression

- Now, we can write

$$\hat{r}_n = \sum_{j=1}^{n} \hat{\beta}_j B_j(x)$$

  where $B_j(x)$ are the basis vectors for the B-splines.
- We follow the pattern of polynomial regression

$$B = \begin{pmatrix} B_1(x_1) & \cdots & B_n(x_1) \\ B_1(x_2) & \cdots & B_n(x_2) \\ \vdots & \vdots & \vdots \\ B_1(x_n) & \cdots & B_n(x_n) \end{pmatrix}$$

Nonparametric Regression

Penalized Regression

- We can rewrite the problem as follows :

  $$\mathrm{argmin}_\beta (Y - B\beta)^T (Y - B\beta) + \lambda\beta^T \Omega\beta$$

  where $B_{ij} = [B_j(X_i)]_{ij}$ and $\Omega_{jk} = \int B_j''(x) B_k''(x) dx$.
- The solution is

  $$\tilde{\beta} = (B^T B + \lambda\Omega)^{-1} B^T Y$$

  and

  $$\hat{Y} = LY$$

  where

  $$L = B(B^T B + \lambda\Omega)^{-1} B^T$$
- We define the effective degree of freedom by $df = \mathrm{trace}(L)$, and we choose λ using the GCV or CV.
- In `R`,

```
out=smooth.spline(x,y,df=10,cv=TRUE)
lines(x,out$y)
```

Nonparametric Regression

Penalized Regression

- In more general case, we will assume that r admits an expansion series wrt to a orthonormal basis $({}_i)_i$ such that

$$r(x) = \sum_{j=1}^{\infty} \beta_j \phi_j(x)$$

- We will approximate r by

$$r_J(x) = \sum_{j=1}^{J} \beta_j \phi_j(x)$$

- The number of terms J will be our smoothing parameter. Our estimate is $\hat{r}(x) = \sum_{j=1}^{J} \hat{\beta}_j \phi_j(x)$.

Nonparametric Regression

Penalized Regression

- The estimate $\hat{\beta}$ is

$$\hat{\beta} = (U^T U)^{-1} U^T Y$$

  where $U = [\phi_j(X_i)]_{ij}$ and $\hat{Y} = SY$ where $S = U(U^T U)^{-1} U^T$
- We can choose J by cross validation.
- Note that $\mathrm{trace}(S) = J$ so the GCV takes the simple form

$$GCV(J) = \frac{SSE}{n} \frac{1}{(1 - J/n)^2}$$

Nonparametric Regression

Penalized Regression : Variance estimation

- Theorem

  Let $\hat{r}_n$ be a linear smoother. Let

  $$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(Y_i - \hat{r}(X_i))^2}{n - 2\nu + \hat{\nu}}$$

  where $\nu = \mathrm{tr}(L)$, $\hat{\nu} = \mathrm{tr}(L^T L)$.
  If r are sufficiently smooth, $\nu = o(n)$ and $\hat{\nu} = o(n)$ then $\hat{\sigma^2}$ is a consistent estimator of σ^2.