Nonparametric Methods I

Wei Li

Syracuse University

Spring 2024

OVERVIEW

Nonparametric regression

Kernel smoothting (Locally weighted averages)

Local linear and polynomial regression

Density estimation

General issues

Nonparametric regression

Nonparametric regression

Given a random pair $(X, Y) \in \mathbb{R}^p \times \mathbb{R}$, the the **regression function** (of Y on X) or **conditional expectation function**:

$$f(x) = E(Y \mid X = x)$$

- The basic goal is to estimate f from some i.i.d. sample $(X_i, Y_i)_{i=1}^n$.

For an i.i.d. sample $(X_i, Y_i) \in \mathbb{R}^p \times \mathbb{R}, i = 1, \ldots, n,,$

$$Y_i = f(X_i) + \epsilon_i, \quad i = 1, \ldots, n$$

- $\epsilon_i, i = 1, \ldots, n$ are i.i.d. random errors, with mean zero and satisfy $E(\epsilon_i | X_i) = 0$.

- nonparametric model does not assume $f(x)$ take a known functional form

example: k-nearest-neighbors regression

$$\hat{f}(x) = \frac{1}{k} \sum_{i \in N_k(x)} y_i$$

where $N_k(x)$ contains the indices of the k closest points of $x_1, \ldots, x_n$ to x.

The regression prediction is a linear smoother (in $\mathbf{y}$):

$$\hat{f}(x) = \sum_{i=1}^{n} w_i(x) y_i = \sum_{i=1}^{n} w\left(x, x_i\right) \cdot y_i = \mathbf{w}(x)\mathbf{y}$$

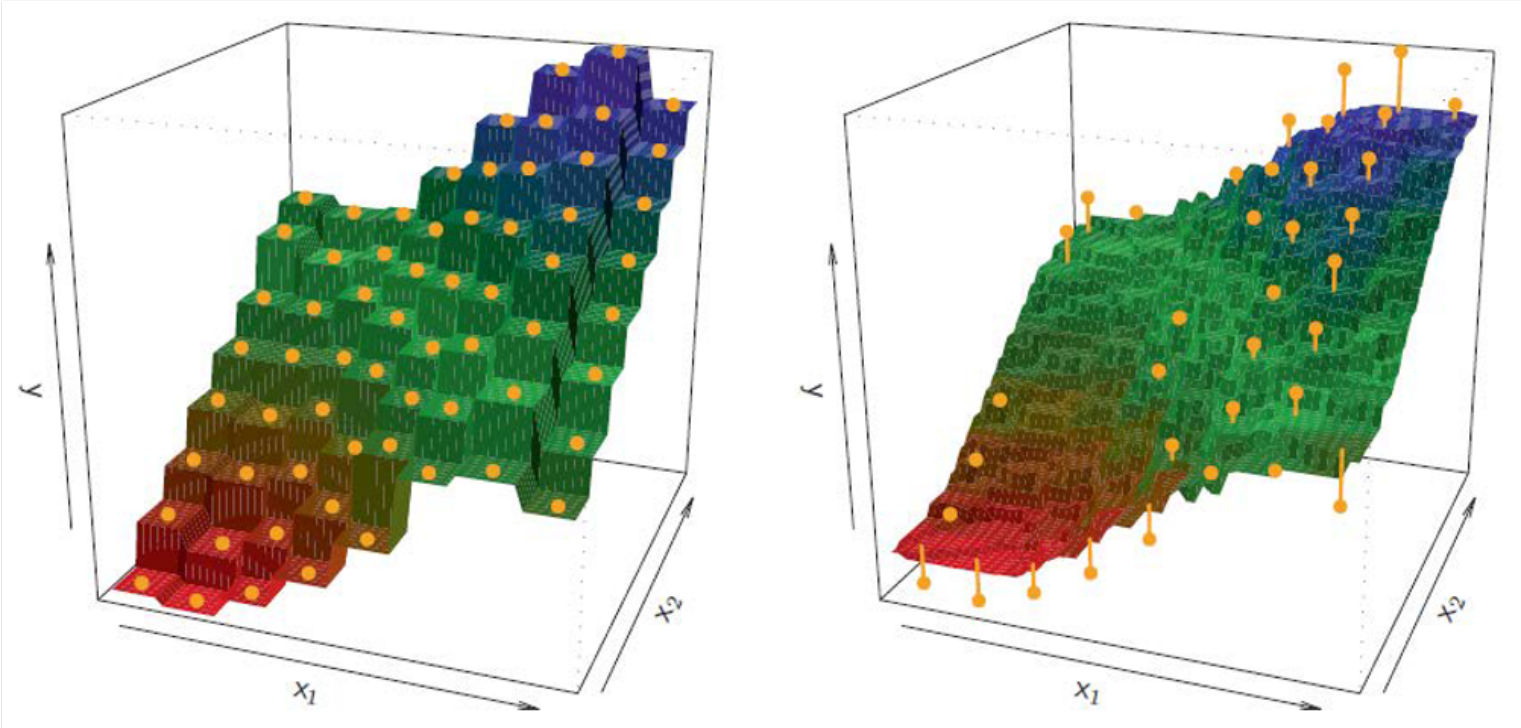
where the weights $w_i(x), i = 1, \ldots, n$ are defined as

$$w_i(x) = w\left(x, x_i\right) = 1(x_i \in N_k(x))/k,$$

Note: KNN is a memory-based model (lazy learner or examplar-based learner).

Curse of dimensionality

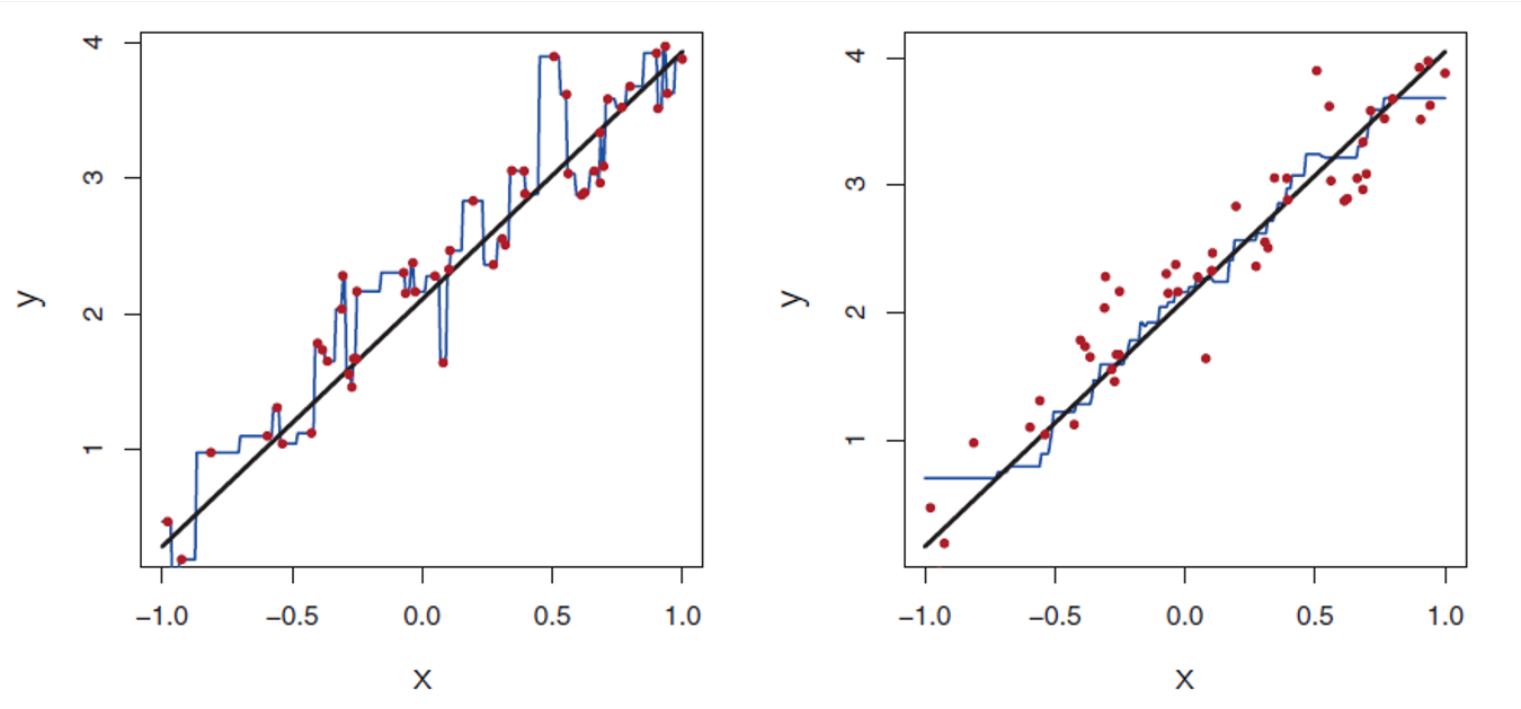
Suppose $x \in \mathbb{R}^p$.



p=2.

- Left: k=1, most flexible fit, high variation
- Right: k=9, smoother fit, less variation

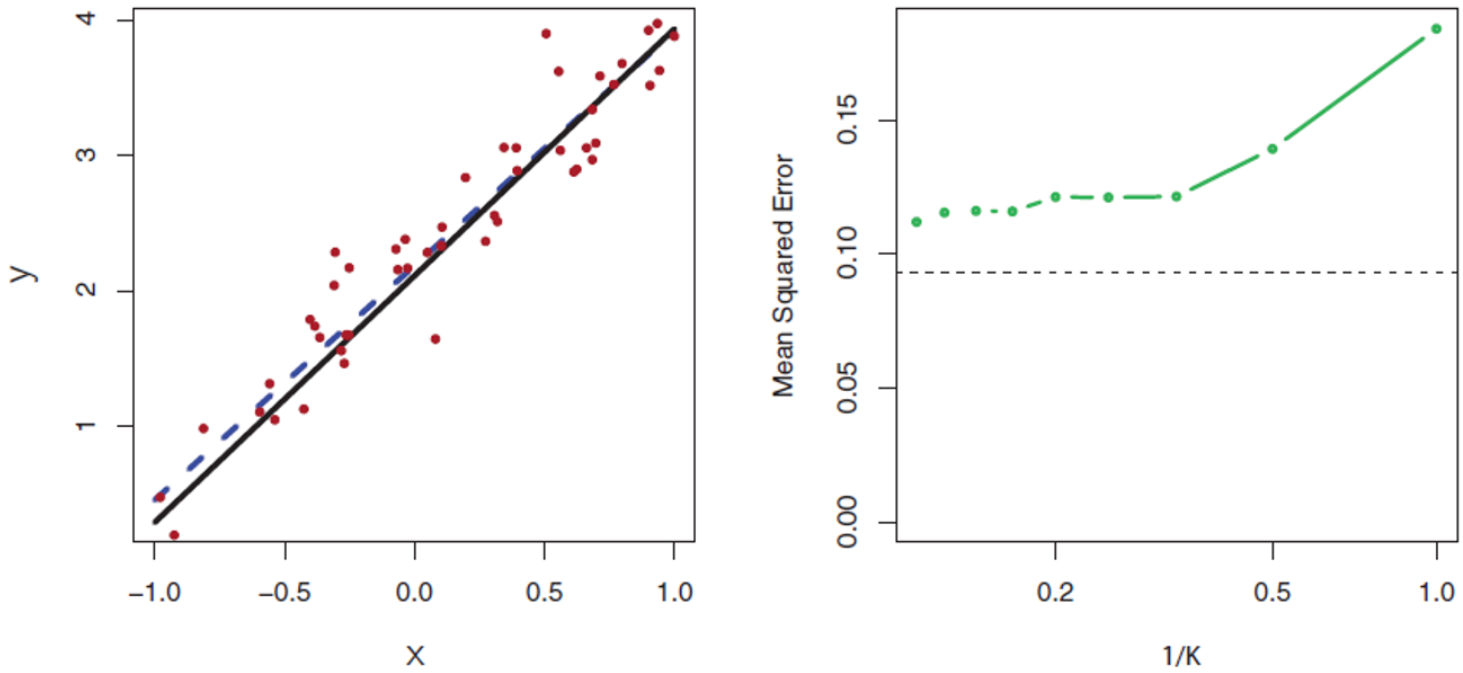
Source of Image: ISL Chapter 3.



p=1, truth=linear

- left: k=1
- right: k=9

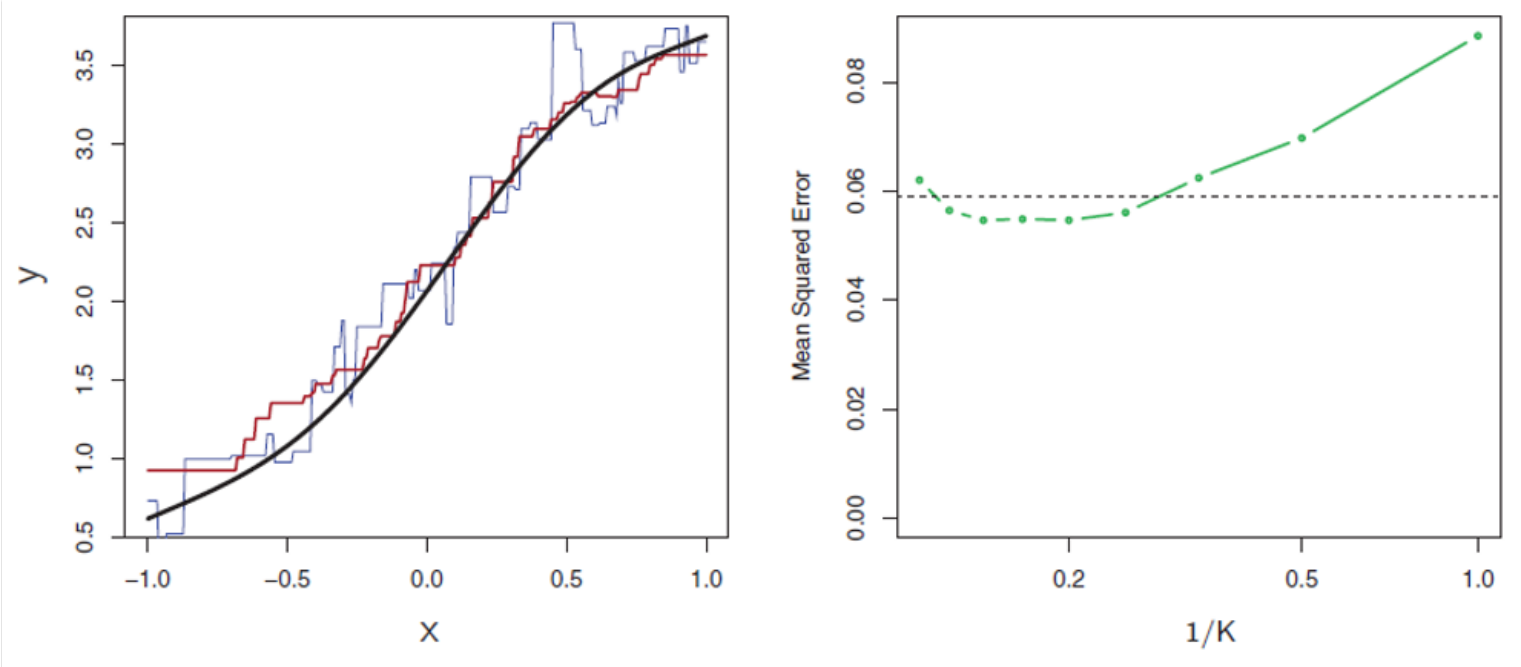
Source of Image: ISL Chapter 3.



p=1, truth=linear.

- Left: black(truth), dashed(LS fit)
- Right: dashed black (LS fit test MSE), dashed green (KNN test MSE)

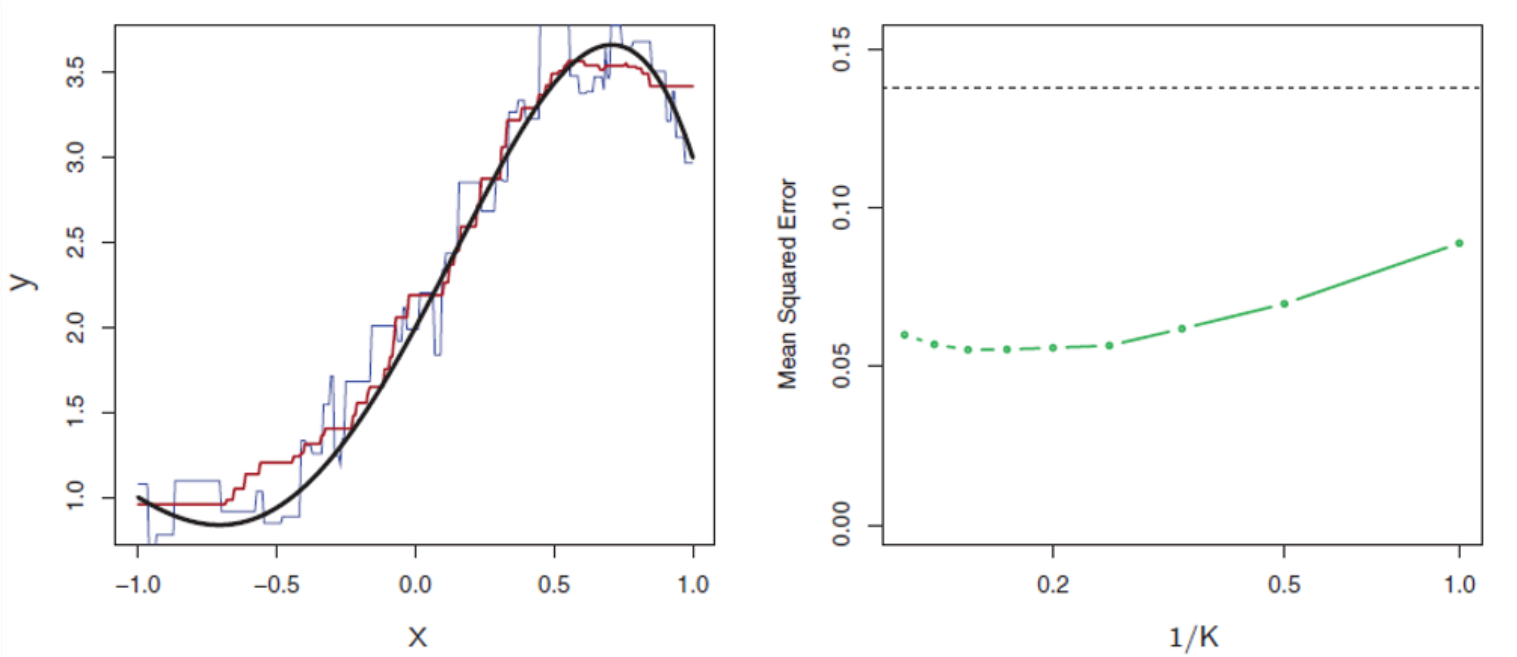
Source of Image: ISL Chapter 3.



p=1, truth=nonlinear.

- Left: black(truth), blue(k=1), red(k=9)
- Right: dashed black (LS fit test MSE), dashed green (KNN test MSE)

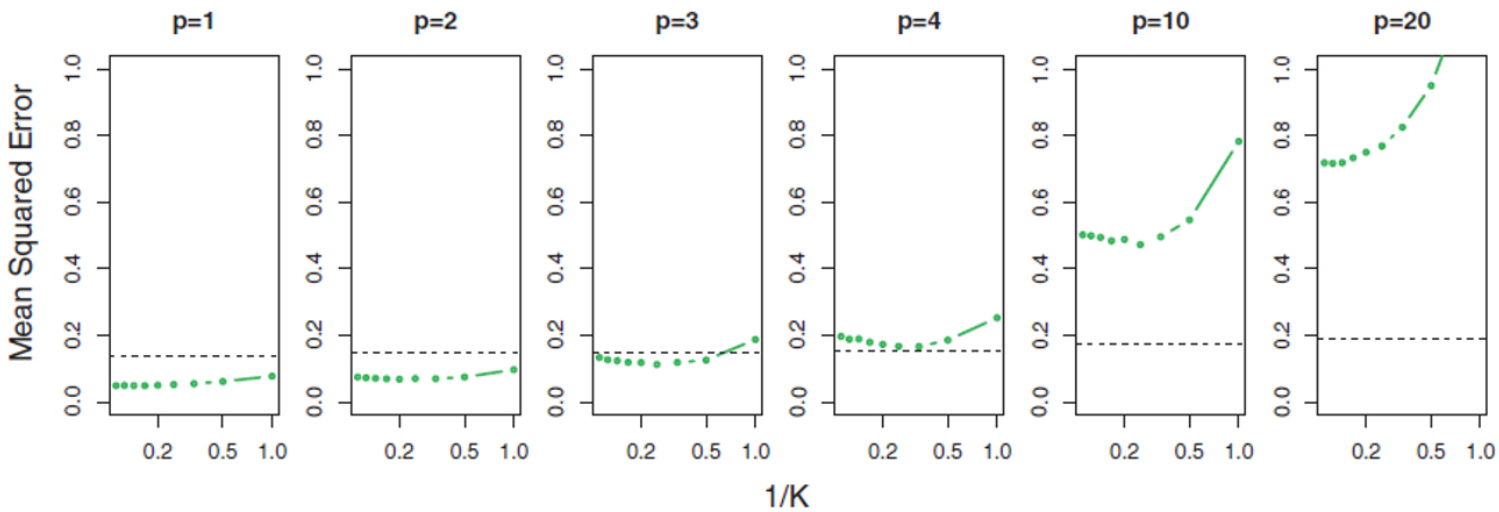
Source of Image: ISL Chapter 3.



p=1, truth=nonlinear.

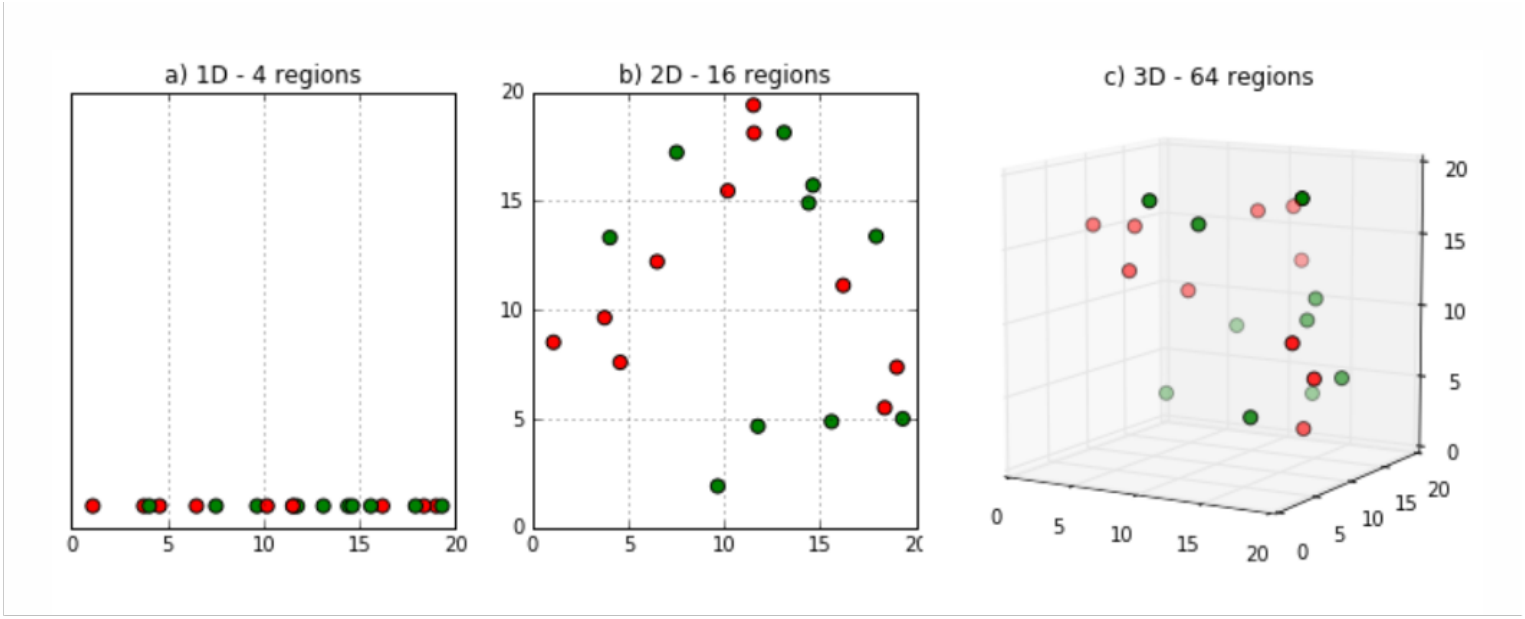
- Left: black(truth), blue(k=1), red(k=9)
- Right: dashed black (LS fit test MSE), dashed green (KNN test MSE)

Source of Image: ISL Chapter 3.



If p is greater than 4, linear regression is superior to KNN even for nonlinear truth (sample size fixed)

Source of Image: ISL Chapter 3.



Curse of Dimensionality

The **curse of dimensionality** refers to the principle that nonparametric estimation may become exponentially harder as the number of dimensions increases (with the exception of additional structure).

Source of Image:
https://deepai.org/machine-learning-glossary-and-terms/curse-of-dimensionality

Convergence rates (k-NN regression)

$$\begin{aligned}\mathrm{MSE}\left(\boldsymbol{x}_0\right) &= \mathrm{E}_{\boldsymbol{\tau}}\left[\hat{f}\left(\boldsymbol{x}_0\right) - f\left(\boldsymbol{x}_0\right)\right]^2 \\ &= \mathrm{Var}\left(\frac{1}{k}\sum_{i\in N_k(\boldsymbol{x}_0)} y_i\right) + \left(f\left(x_0\right) - E\left(\frac{1}{k}\sum_{i\in N_k(\boldsymbol{x}_0)} y_i\right)\right)^2\end{aligned}$$

Assume f Lipschitz continuous

- variance term $\approx \sigma^2/k$
- squared bias term $\approx (k/n)^{2/p}$

With $k \asymp n^{2/(2+p)}$, it satisfies $\mathrm{MSE}(\boldsymbol{x}_0) \lesssim n^{-2/(2+p)}$

- optimal rate for estimating a Lipschitz p-variate functions
- requires exponentially more data to achieve a given error

Kernel smoothting (Locally weighted averages)

Kernel smoothting (Locally weighted averages)

Given a **bandwidth** $h > 0$, the **Nadaraya-Watson** kernel regression estimate is defined as

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{n} K_h(x_0, x_i)\, y_i}{\sum_{i=1}^{n} K_h(x_0, x_i)}$$

Kernel smoothing is a linear smoothing method, with weights

$$w(x_0, x_i) = K_h(x_0, x_i) / \sum_{j=1}^{n} K_h(x_0, x_j).$$

That is $\hat{f}(x_0) = \sum_{i=1}^{n} w(x_0, x_i) y_i$.

$$\hat{\mathbf{f}} = (\hat{f}(x_1), \ldots, \hat{f}(x_n))^\top = S_h \mathbf{y}$$

where S_h is $n \times n$ matrix and $[S_h]_{i,j} = w(x_i, x_j)$.

In general

$$K_h(x_0, x) = \bar{K}\left(\frac{\|x - x_0\|}{h}\right)$$

for some kernel function $\bar{K} : \mathbb{R} \to \mathbb{R}$ satisfying

$$\int \bar{K}(t)dt = 1, \quad \int t\bar{K}(t)dt = 0, \quad 0 < \int t^2\bar{K}(t)dt < \infty$$

the box-car kernel:

$$\bar{K}(t) = 1(|t| \leq 1/2)$$

the Epanechnikov kernel:

$$\bar{K}(t) = \frac{3}{4}\left(1 - t^2\right)1(|t| \leq 1)$$

the Gaussian kernel:

$$\bar{K}(t) = \frac{1}{\sqrt{2\pi}}\exp\left(-t^2/2\right)$$

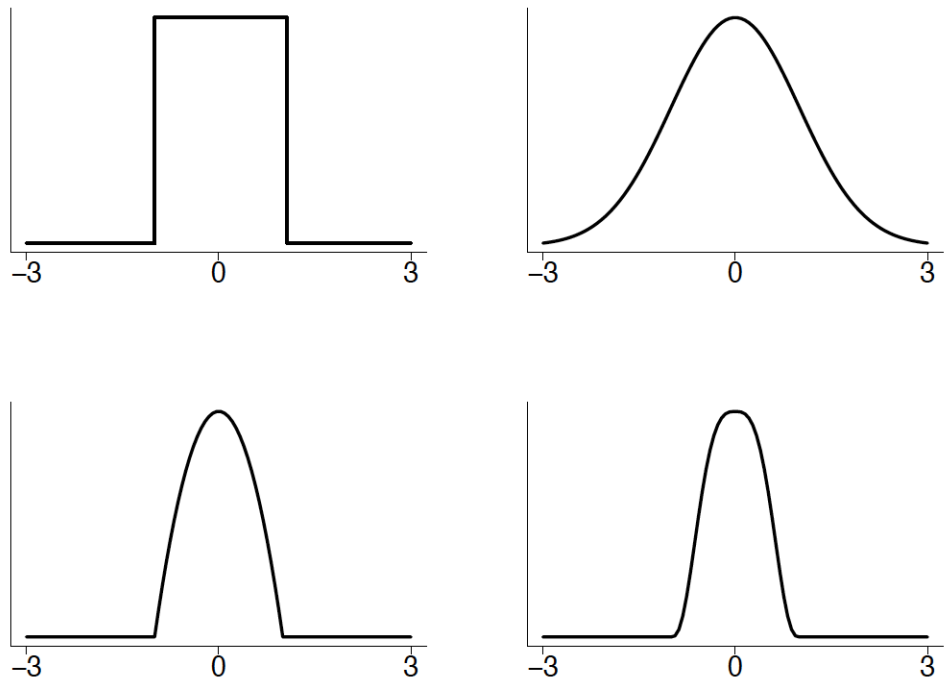
For $p > 1$, the Gaussian kernel can be factored as a product of p Gaussian kernels for each component. Also it would be helpful to standardize the predictors before smoothing.

Boxcar: $\bar{K}(x) = \frac{1}{2} 1\{x : |x| \leq 1\}$

Gaussian: $\bar{K}(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$

Epanechnikov: $\bar{K}(x) = \frac{3}{4}\left(1 - x^2\right) 1\{x : |x| \leq 1\}$

Tricube: $\bar{K}(x) = \frac{70}{81}\left(1 - |x|^3\right)^3 1\{x : |x| \leq 1\}$



boxcar (top left), Gaussian (top right), Epanechnikov (bottom left), and tricube (bottom right)

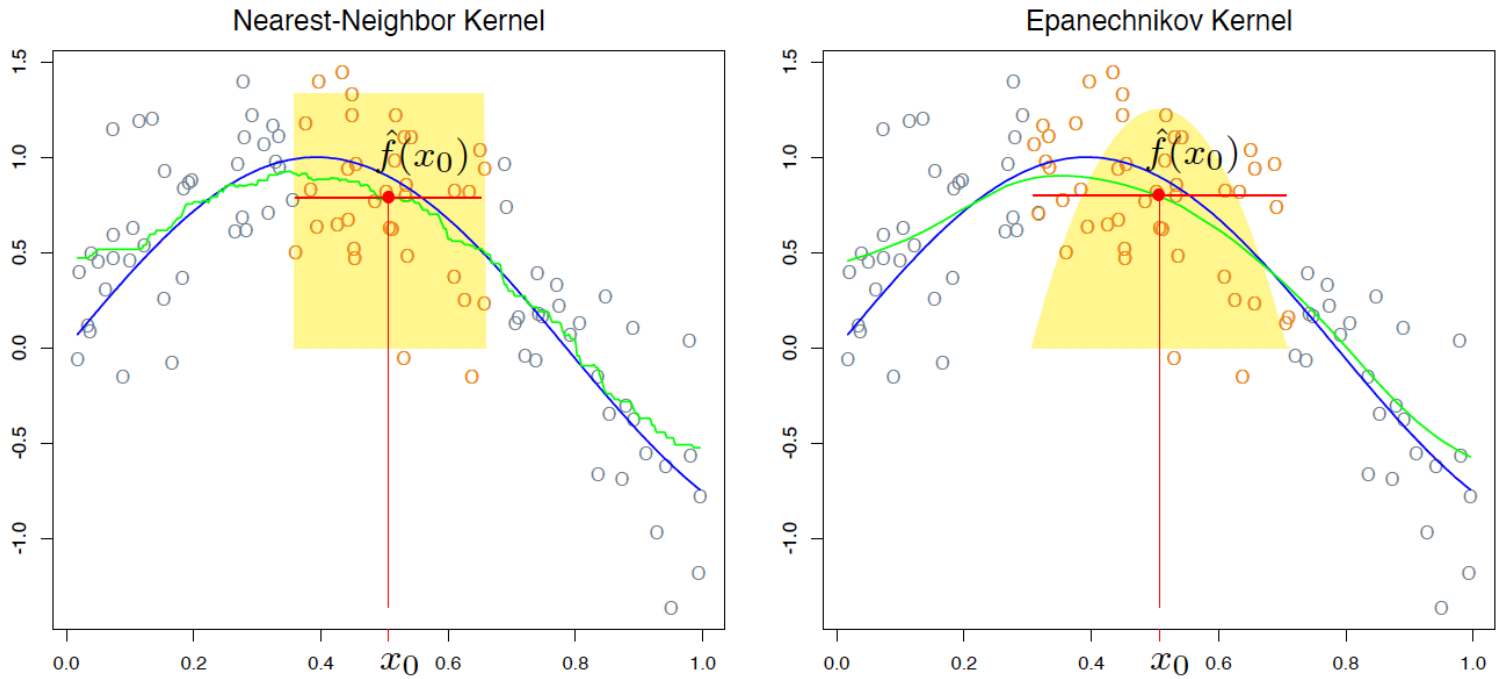
special case

The k-nearest-neighbor estimator is just a raw (discontinuous) moving average of neighboring response with

$$K_h(x_0, x) = \bar{K}\Big(\frac{\|x - x_0\|}{\|x_{(k)} - x_0\|}\Big),$$

where

$$\bar{K}(t) = \begin{cases} 1 & |t| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$



The nearest-neighbor kernel compared with Epanechnikov kernel.

- blue curve is truth
- the red point is $\hat{f}(x_0)$
- red circles are those observations contributing to the fit at x_0
- solid yellow region indicates the weights assigned to observations.

ESL Figure 6.1

alternative form

An alternative form of $K_h(x_0, x)$:

$$K_h(x_0, x) = \bar{K}\left(\frac{x_{01} - x_1}{h_1}\right) \times \ldots \times \bar{K}\left(\frac{x_{0p} - x_p}{h_p}\right)$$

Then

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{n} \bar{K}\left(\frac{x_{01} - x_{i1}}{h_1}\right) \times \ldots \times \bar{K}\left(\frac{x_{0p} - x_{ip}}{h_p}\right) y_i}{\sum_{i=1}^{n} \bar{K}\left(\frac{x_{01} - x_{i1}}{h_1}\right) \times \ldots \times \bar{K}\left(\frac{x_{0p} - x_{ip}}{h_p}\right)}$$

Different bandwidth could have been used to offset the unit effects in predictors.

Convergence rates (kernel regression)

In theory, for $X \in \mathbb{R}^p$ and f is s-times differentiable ($s \geq 1$), the minimal achievable rate of MSE by the best nonparametric method is $n^{-2s/(2s+p)}$.

- Suppose f is at least two-times differentiable ($s = 2$), and X has a non-zero, differentiable density and the support is unbounded, using the **Gaussian kernel** (*second order kernel*), one can show

$$\text{Bias}(\hat{f}(x)) \leq \tilde{C}_1 h^2$$
$$\text{Var}(\hat{f}(x)) \leq \frac{\tilde{C}_2}{nh^p}$$

As long as $h \to 0$ and $nh^p \to \infty$, the MSE goes to 0.

Balancing squared bias error and variance, the optimal choice is $h = n^{-1/(p+4)}$ and the MSE is $n^{-4/(p+4)}$.

- the rate is "saturated" (bias rate of order not improve with $s \geq 2$).
- poor bias rate near the boundaries if estimated on a bounded support

Is it possible to use a kernel estimate to obtain the optimal rate when $s > 2$? The answer is yes, with the higher order kernel function $\bar{K}$.

A kernel function $\bar{K}$ is said to be of order k provided that

$$\int \bar{K}(t)dt = 1, \int t^j \bar{K}(t)dt = 0, j = 1, \ldots, k-1, \text{ and } \quad 0 < \int t^k \bar{K}(t)dt < \infty$$

- Previous kernels were of order 2.
- An example of a 4 th-order kernel: $\bar{K}(t) = \frac{3}{8}\left(3 - 5t^2\right) 1\{|t| \leq 1\}$.
  - it can take negative values (a ubiquitous feature of higher-order kernel)

The second issue with the Gaussian kernel is related to the order $O\left(h^2\right)$ of bias (assuming unbounded support)

- When estimated at the boundary points of the data, the bias becomes $O(h)$. This is the so-called "boundary effects".
- One possible solution is to use some **boundary-corrected kernel**.

Perhaps better use the local polynomial regression, or series-function-based regression.

General rates results

For kernel-based estimator, when f is s-times continuously differentiable (H"{o}lder-s-smoothness) and the *higher-order* kernel function is used, then

$$\sup_x |\hat{f}(x) - f(x)| = O(h^s) + O_p\left(\sqrt{\frac{\log n}{nh^p}}\right)$$

Balancing the two parts, the optimal choice $h \asymp (\log n/n)^{1/(2s+p)}$, yielding the sup-norm optimal rate $(\log n/n)^{s/(2s+p)}$.

- When $h < (\log n/n)^{1/(2s+p)}$, we call it **under-smoothing** (relative to the optimal rate)–variance dominant.
- When $h > (\log n/n)^{1/(2s+p)}$, we call it **over-smoothing** (relative to the optimal rate)–bias dominant.

under-smoothing/over-smoothing = overfitting/underfitting in asymptotic sense

Bandwidth selection

1. rule of thumb: $\hat{h} = \hat{c} \times (1/n)^{1/(4+p)}$ i.e., $s = 2$, with estimated const.
2. cross-validation:

Choose from a sequence say, $\{h_1, \ldots, h_m\}$; randomly split the data into a K-parts partition $(C_1, \ldots, C_K)$

- For each part k:
  - For each value of h, fit the regression using the remaining $K - 1$ parts (excluding k-th part), and denote the solution by $\hat{f}_h^{-(k)}$.
  - For each value of h, compute the prediction error on the k-th part,

$$CV_k(h) = \frac{1}{n_k} \sum_{i \in C_k} \left(y_i - \hat{f}_h^{-(k)}(x_i) \right)^2$$

- Take the average $CV(h) = \frac{1}{K} \sum_{k=1}^{K} CV_k(h)$
- Find the best parameter $\hat{h}$ which minimizes $CV(h)$.

One may repeat the whole process for a couple of times, obtaining $CV(h)$'s for different partitions, then take another average as the final CV estimate; and select h based on this final estimate.

Inference about kernel estimation

For the kernel estimator $\hat{f}$ (using Gaussian kernel)

$$\hat{f}(x) - f(x) = (E\hat{f}(x) - f(x)) + (\hat{f}(x) - E\hat{f}(x))$$

- $B_n(x) = E\hat{f}(x) - f(x) = O\left(h^2\right)$ is the bias
- $\xi_n(x) = \hat{f}(x) - E\hat{f}(x) = O_p\left(\sqrt{1/nh^p}\right)$ is the stochastic variation and $\sqrt{nh^p}\xi_n(x) \to N\left(0, V_n(x)\right)$ for some variance $V_n(x)$.

A valid pointwise confidence interval for $f(x)$ is

$$\left(\hat{f}(x) - B_n(x) - z_{\alpha/2}\sqrt{\frac{V_n(x)}{nh^p}}, \hat{f}(x) - B_n(x) + z_{\alpha/2}\sqrt{\frac{V_n(x)}{nh^p}}\right)$$

where $z_{\alpha/2}$ is the upper $\alpha/2$ quantile of a standard normal distribution.

Inference about kernel estimation

$$\left(\hat{f}(x) - B_n(x) - z_{\alpha/2}\sqrt{\frac{V_n(x)}{nh^p}}, \hat{f}(x) - B_n(x) + z_{\alpha/2}\sqrt{\frac{V_n(x)}{nh^p}}\right)$$

To deal with the bias B_n:

- *undersmooth* the kernel estimate, say $h = o\left((nh^p)^{-1/4}\right)$, i.e., $h = o(n^{-1/(4+p)})$
- or estimate B_n

Without undersmoothing, ignoring B_n will in general produce a biased pointwise confidence interval

*Since the kernel regression is also linear smoother, can use the formula for the pointwise CI for series estimation.

Confidence bands

It is possible to obtain (sup-norm) confidence band (say through bootstrap). Suppose R_α is the upper α quantile of $\sup_x |\hat{f}(x) - f(x)|$, then

$$P\left(f(x) \in \left(\hat{f}(x) - R_\alpha, \hat{f}(x) + R_\alpha\right), \forall x\right) = 1 - \alpha.$$

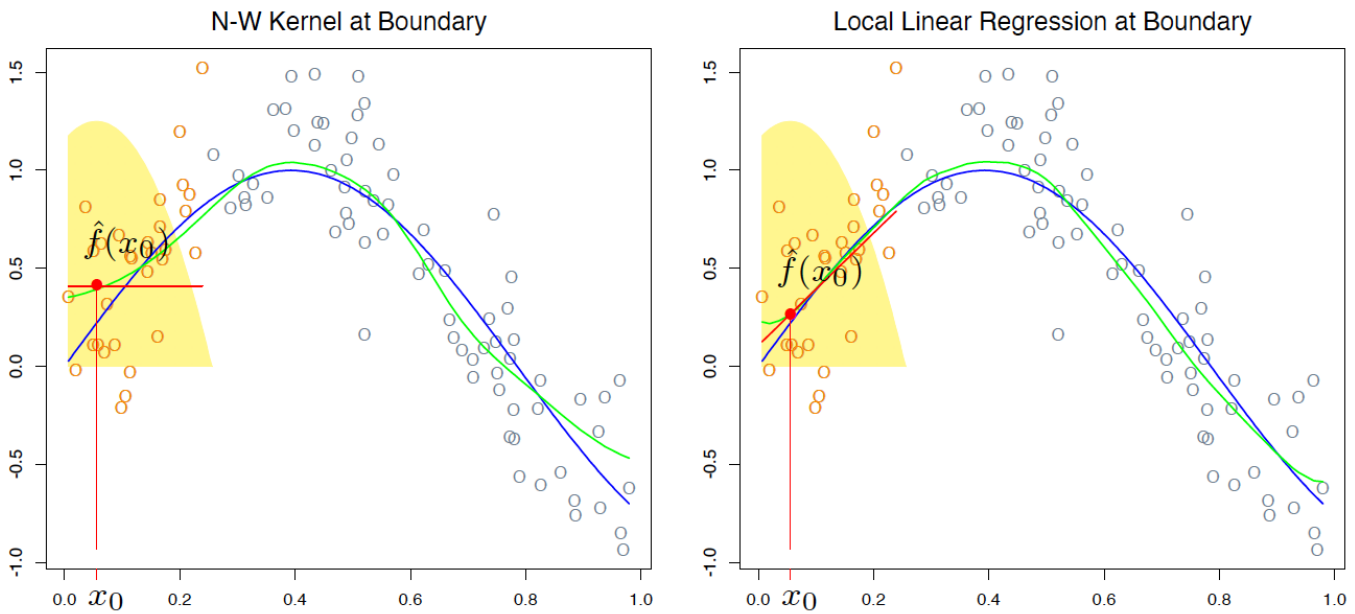
So a $1-\alpha$ uniform confidence band for $f(x)$ is $\left(\hat{f}(x) - R_\alpha, \hat{f}(x) + R_\alpha\right)$.

One can use bootstrap for $\sup_x |\hat{f}(x) - \mathrm{E}(\hat{f}(x))|$ to address the variance estimate, using debiased estimator or undersmoothing to address the bias term. See future lessons.

Local linear and polynomial regression

Local linear and polynomial regression

The kernel regression suffers from poor bias at the boundaries of the domain of the inputs $x_1, \ldots x_n$.



NW estimator compared with local linear regression.

ESL Figure 6.1

Locally linear (locally weighted) regression: $p = 1$

At each point, we make prediction by using linear regression weighting only nearby points. At each target point x_0, solve

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{n} K_h\left(x_0, x_i\right)\left(y_i - \alpha\left(x_0\right) - \beta\left(x_0\right) x_i\right)^2$$

- vector-valued function $b(x)^\top = (1, x)$.
- $\mathbf{B}$ be the $n \times 2$ regression matrix with ith row $b\left(x_i\right)^\top$,
- $\mathbf{W}\left(x_0\right)$ the $n \times n$ diagonal matrix with i th diagonal element $K_h\left(x_0, x_i\right)$

$$\begin{aligned}\hat{f}\left(x_0\right) &= b\left(x_0\right)^\top \left(\mathbf{B}^\top \mathbf{W}\left(x_0\right) \mathbf{B}\right)^{-1} \mathbf{B}^\top \mathbf{W}\left(x_0\right) \mathbf{y} \\ &= \sum_{i=1}^{n} l_i\left(x_0\right) y_i = l(x_0)^\top \mathbf{y}\end{aligned}$$

A local polynomial of any degree d:

$$\min_{\alpha(x_0),\beta_j(x_0),j=1,\ldots,d} \sum_{i=1}^{n} K_h\left(x_0, x_i\right)\left(y_i - \alpha\left(x_0\right) - \sum_{j=1}^{d} \beta_j\left(x_0\right) x_i^j\right)^2$$

with solution $\hat{f}\left(x_0\right) = \hat{\alpha}\left(x_0\right) + \sum_{j=1}^{d} \hat{\beta}_j\left(x_0\right) x_0^j$.

$$\hat{f}(x_0) = b(x_0)\left(\mathbf{B}^\top \mathbf{W}\left(x_0\right)\mathbf{B}\right)^{-1}\mathbf{B}^\top \mathbf{W}(x_0)\mathbf{y} = l(x_0)^\top \mathbf{y}$$

- $b(x) = \left(1, x, \ldots, x^d\right)$
- $\mathbf{B}$ is an $n \times (d+1)$ matrix with i-th row $b(x_i) = \left(1, x_i, \ldots, x_i^d\right)$,
- $\mathbf{W}(x_0)$ is as before.

Note local polynomial regression is a linear smoother, so $\hat{f} = S_h\mathbf{y}$ where $[S_h]_{i,j} = l_j(x_i)$.

p>1

Let $b(X)$ be a vector of polynomial terms in X of maximum degree d.

- For example, with $d = 1$ and $p = 2$ we get $b(X) = (1, X_1, X_2)$; with $d = 2$ we get $b(X) = \left(1, X_1, X_2, X_1^2, X_2^2, X_1 X_2\right)$

At each $x_0 \in \mathbb{R}^p$ solve

$$\min_{\beta(x_0)} \sum_{i=1}^{n} K_h\left(x_0, x_i\right) \left(y_i - b\left(x_i\right)^\top \beta\left(x_0\right)\right)^2$$

to produce the fit $\hat{f}\left(x_0\right) = b\left(x_0\right)^\top \hat{\beta}\left(x_0\right)$.

The polynomials terms are constructed as tensor products of basis functions. See series estimators.

Convergence rates

- The convergence rates of local polynomial regression with (correct) *odd* degree used are similar to the kernel regression with higher-order kernel.
  - with optimal bandwidth $h \asymp (1/n)^{1/(2s+p)}$
- It can remove boundary bias and design bias (arising from the underlying density of X), thus better rates in general compared with kernel regression with second-order kernel, and in particular near the boundaries.

Local likelihood

Associated with each observation y_i is a parameter $\theta_i = \theta(x_i) = x_i^\top \beta$ linear in the covariate(s) x_i, and inference for β is based on the loglikelihood $\ell(\beta) = \sum_{i=1}^n \log L\left(y_i, x_i^\top \beta\right)$.

Model $\theta(x)$ by using the likelihood local to x_0 for inference of $\theta(x_0) = x_0^\top \beta(x_0)$:

$$\ell\left(\beta(x_0)\right) = \sum_{i=1}^n K_h(x_0, x_i) \log L\left(y_i, x_i^\top \beta(x_0)\right)$$

Local version of multi-class linear logistic regression model

The data consist of features x_i and an associated categorical response $y_i \in \{1, 2, \ldots, J\}$, and the linear model has the form

$$\Pr(Y = j \mid X = x) = \frac{e^{\beta_{j0} + \beta_j^\top x}}{1 + \sum_{k=1}^{J-1} e^{\beta_{k0} + \beta_k^\top x}}$$

where $\beta_{J0} := 0$ and $\beta_J := 0$ by the definition of the model.

The local log-likelihood for this J class model can be written

$$\begin{array}{l} \sum_{i=1}^{n} K_h(x_0, x_i) \left\{ \beta_{y_i 0}(x_0) + \beta_{y_i}(x_0)^\top (x_i - x_0) \right. \\ \left. - \log\left(1 + \sum_{k=1}^{J-1} \exp\left(\beta_{k0}(x_0) + \beta_k(x_0)^\top (x_i - x_0)\right)\right) \right\} \end{array}$$

Density estimation

Density estimation

- Data $X_1, \ldots, X_n$ are contained in the unit cube $\mathcal{X} = [0,1]^p$.
- Divide $\mathcal{X}$ into bins, or sub-cubes, of length h.
- There are $M \approx (1/h)^p$ such bins and each has volume h^p.
- Denote the bins by $B_1, \ldots, B_M$.

1. Assuming the density estimate should be constant in each cube.
2. Letting that constant value be proportional to the number of observations falling in the cube

A heuristic estimator for a given point $x \in B_j$:

$$\widehat{p}_n(x) = \frac{\text{number of observations within } B_j}{n} \times \frac{1}{\text{volume of the bin}}$$

For a given point $x \in B_j$:

$$\widehat{p}_n(x) = \frac{\text{number of observations within } B_j}{n} \times \frac{1}{\text{volume of the bin}}$$

The **histogram density estimator** is

$$\widehat{p}_h(x) = \frac{1}{nh^p} \sum_{j=1}^{M} \#\{i : X_i \in B_j\} \times 1\,(x \in B_j)$$

A variant is to allow the fraction of data points computed around the evaluation point x. e.g.: boxcar $p = 1$

$$\widehat{p}_h(x) = \frac{1}{2nh} \sum_{i=1}^{n} 1\{|X_i - x| \leq h\} = \frac{1}{2nh} \#\{i : |X_i - x| \leq h\}$$

KDE (kernel density estimate) or Parzen estimate

Suppose $p \geq 1$. The smooth **KDE** or **Parzen** estimate is

$$\hat{p}_h(x) = \frac{1}{nh^p} \sum_{i=1}^{n} K_h(x, x_i)$$

Here, $K_h(x, y) \propto \bar{K}(\|x - y\|/h)$ (as radial basis kernel) for some kernel function $\bar{K}$.

The kernel is assumed to satisfy

- $\int \bar{K}(x)dx = 1$, $\int x\bar{K}(x)dx = 0$
- $\sigma_{\bar{K}}^2 \equiv \int x^2 \bar{K}(x)dx > 0$.

Note that different h's may be used in different coordinates, so

$$\hat{p}_h(x) = \frac{1}{nh_1 \dots h_p} \sum_{i=1}^{n} K_h(x, x_i) = \frac{1}{nh_1 \dots h_p} \sum_{i=1}^{n} \prod_{j=1}^{p} \bar{K}\Big(\frac{|x_j - x_{ij}|}{h_j}\Big)$$

$K_h(x, y) = \prod_{j=1}^{p} \bar{K}(|x_j - y_j|/h_j)$.

When $K_h(x, y) \propto \bar{K}(\|x - y\|/h)$ for some kernel function $\bar{K}$,

e.g. Gaussian

$$\hat{p}_h(x) = \frac{1}{nh^p} \sum_{i=1}^{n} K_h(x, x_i) = \frac{1}{nh^p (2\pi)^{p/2}} \sum_{i=1}^{n} \exp\Big(- \frac{\|x_i - x\|^2}{2h^2} \Big)$$

Remark

The choice of bandwidth follows similar approaches of kernel regression

1. rule of thumb: $\hat{h} = \hat{c} \times (1/n)^{1/(4+p)}$ (assuming $s = 2$, i.e., f twice continuosly differentiable), with estimated const.
2. use general cross-validation or a tailored version of it.

The convergence rate and inferential procedure for kernel density estimation is similar to that of kernel regression.

The density estimate enjoys the benefits of having an explicit likelihood or density function, this is in contrast with the **generative approach** by which artificial data can be generated which resembles the empirical distribution and yet does not rely on any explicit likelihood or density estimate.

General issues

General issues

- The bandwidth h need not be the same for all dimensions
  - may be separately chosen for different component/coordinate
- The choice of bandwidth requires belief or knowledge about the underlying degree of smoothness s of the function f (so does the convergence rate), it is desired to have methods that adapt to the degree of smoothness automatically
  - **adaptive estimation** (general theme in the state of art)
- The curse of dimensionality is ubiquitous for nonparametric methods
  - general requires much larger sample size (relative to the dimension) to be successful
  - The curse of dimensionality may be alleviated if some further structure can be reasonably assumed on the function f and with methods that take into account this structure (general theme in the state of art)
- The other class of nonparametric methods (**series estimator** or **sieve estimators**) to is not free of these challenges.