

修勻 (Graduation)

政治大學統計系余清祥

2005年11月21日

第十週：估計密度函數

<http://csyue.nccu.edu.tw>

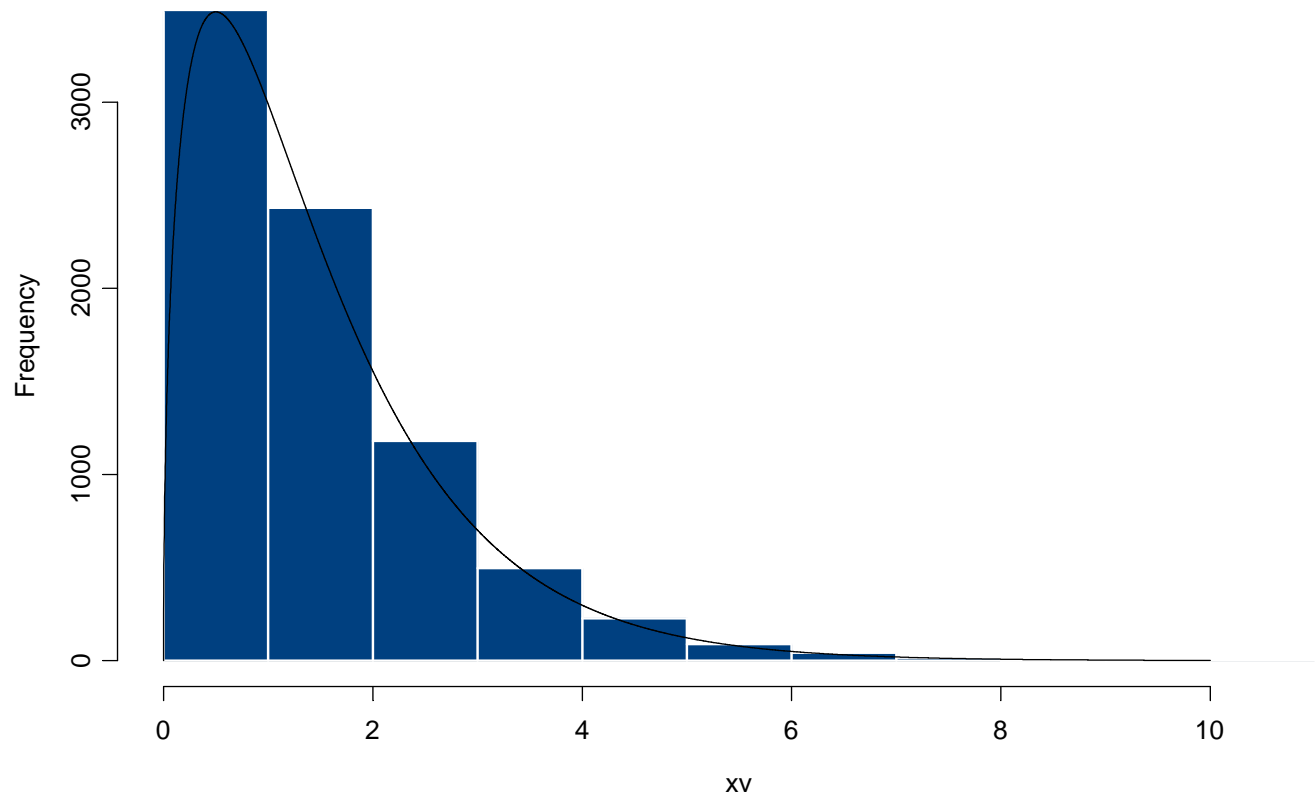




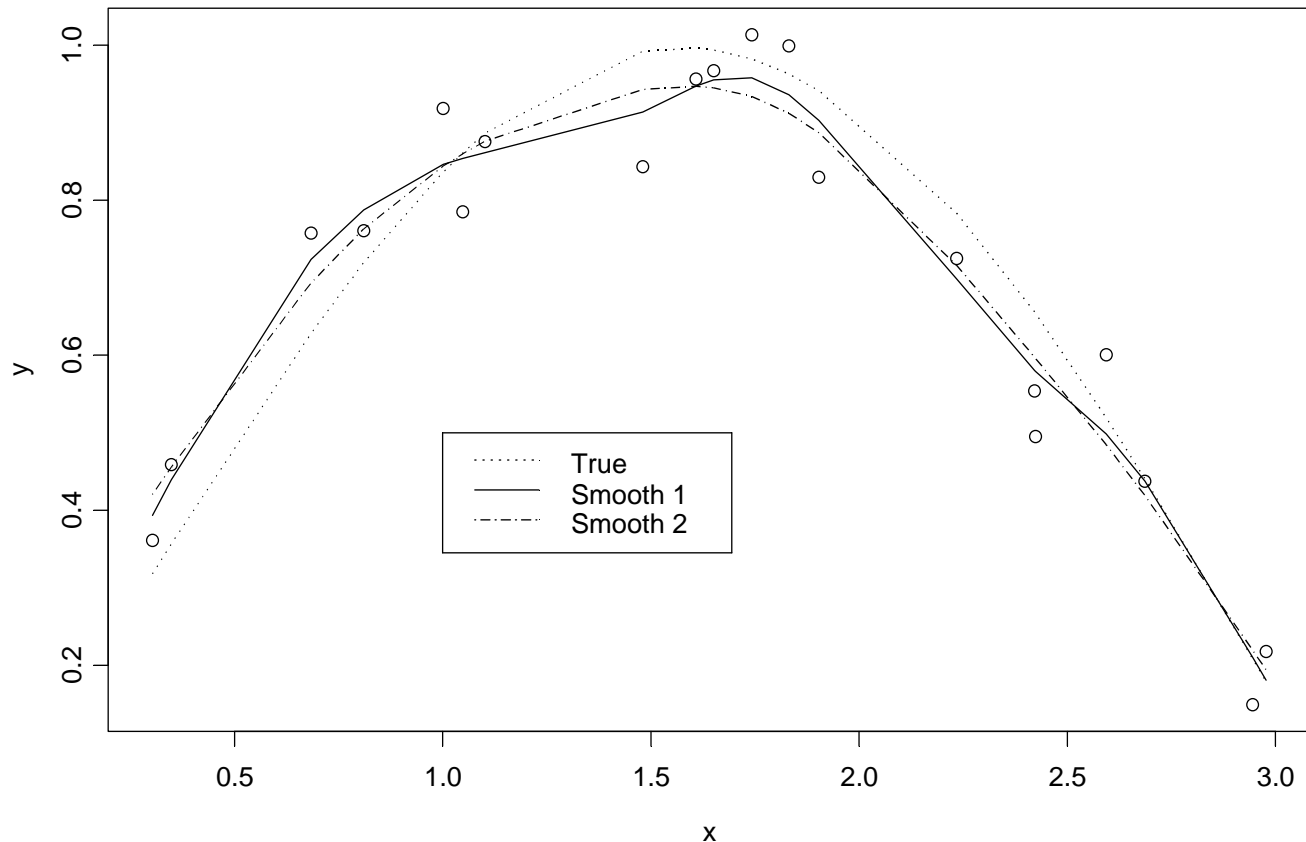
Density Estimation

- Estimate the density functions without the assumption that the p.d.f. has a particular form.
- The idea of estimating c.d.f. (i.e., $F(x_0)$) is to count the number of observations not larger than x_0 . Since the p.d.f. is the derivative of c.d.f., the natural estimate of the p.d.f. would be $\hat{f}(x_0) = \sum I\{x_i = x_0\} / n$. However, this is likely not a good estimate since most points have zero density.

Therefore, we may want to assign a nonzero weight to points near points with observations. Intuitively, the weight should be larger if a point is close to an observation, but this is not necessary to be true.



- Smoothing, a process of obtaining a corresponding smooth set of values from irregular set of observed values, is closely linked computationally to density estimation.





■ Histogram

- The histogram is the simplest and most familiar method of a density estimator.
- Break the interval $[a, b]$ into m bins of equal width h , say $a = a_0 < a_1 < \dots < a_{m-1} < a_m = b$. Then the density estimate of $x \in [a, b]$ is

$$\hat{f}(x) = \frac{1}{n} \sum_{j=1}^m \frac{n_j}{h} \cdot I\{x \in [a_{j-1}, a_j]\},$$

where n_j is the number of observations in the interval $x \in [a_{j-1}, a_j]$.



Notes:

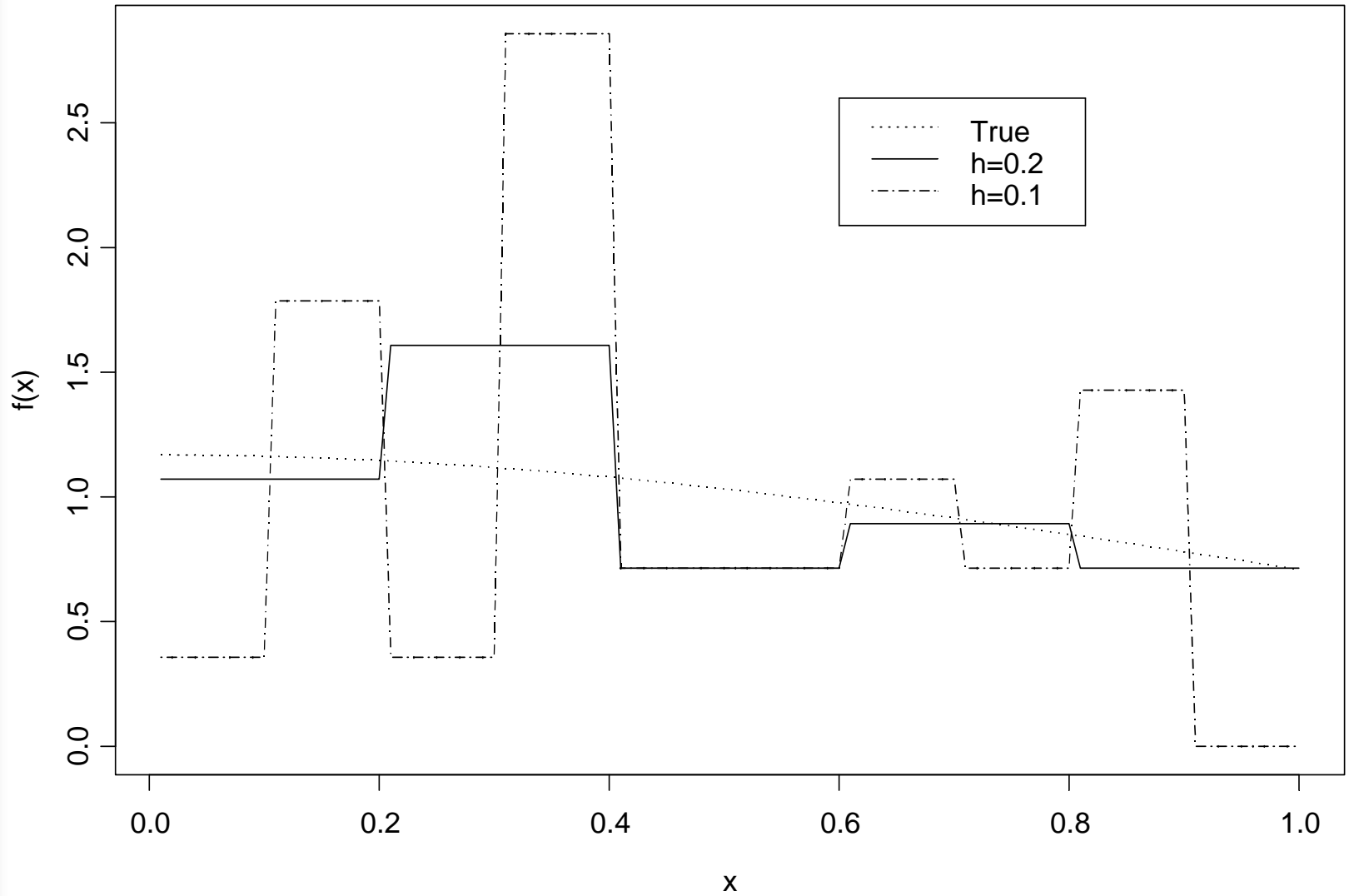
(1) The histogram density estimate looks like the sample c.d.f. and is a step function.

(2) The smaller h is, the smoother the density estimate will be. However, given a finite number of observations, when h is smaller than most of the distances between two points, the density estimate would become more discrete.

Q: What is the “optimal” choice of h ?

(The number of bins for drawing a histogram)

Histogram Estimate (n=28, N(0,1))





- The Naïve Density Estimator

→ Instead of rectangle, allow the weight is centered on x . From Silverman (1986),

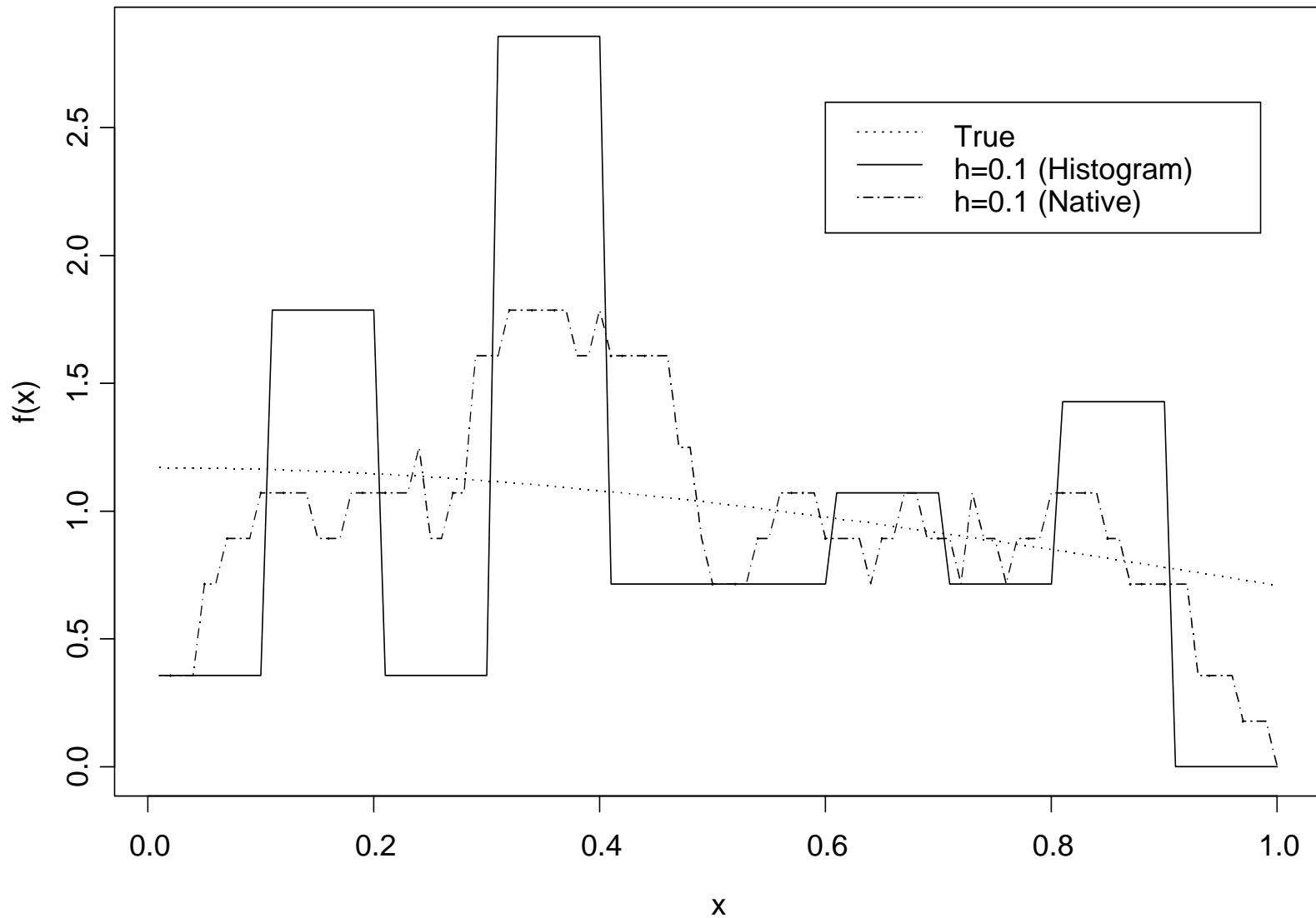
$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} w\left(\frac{x - x_i}{h}\right),$$

where

$$w(x) = \begin{cases} \frac{1}{2}, & |x| < 1; \\ 0, & \text{Otherwise.} \end{cases}$$

Because the estimate is constructed from a moving window of width $2h$, it is also called a *moving-window histogram*.

Histogram Estimate (n=28, N(0,1))





- Kernel Estimator:

→ The naïve estimator is better than the histogram, since weight is based on distance between observations and x . However, it also has jumps (similar to the histogram estimate) at the observation points. By modifying the weight function $w(\cdot)$ to be more continuous, the raggedness of the naïve estimate can be overcome.



→ The kernel estimate is as following:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x - x_i}{h}\right),$$

where $\int_{-\infty}^{\infty} K(t)dt = 1$ is the kernel of the estimator.

→ Usual choices of kernel functions:

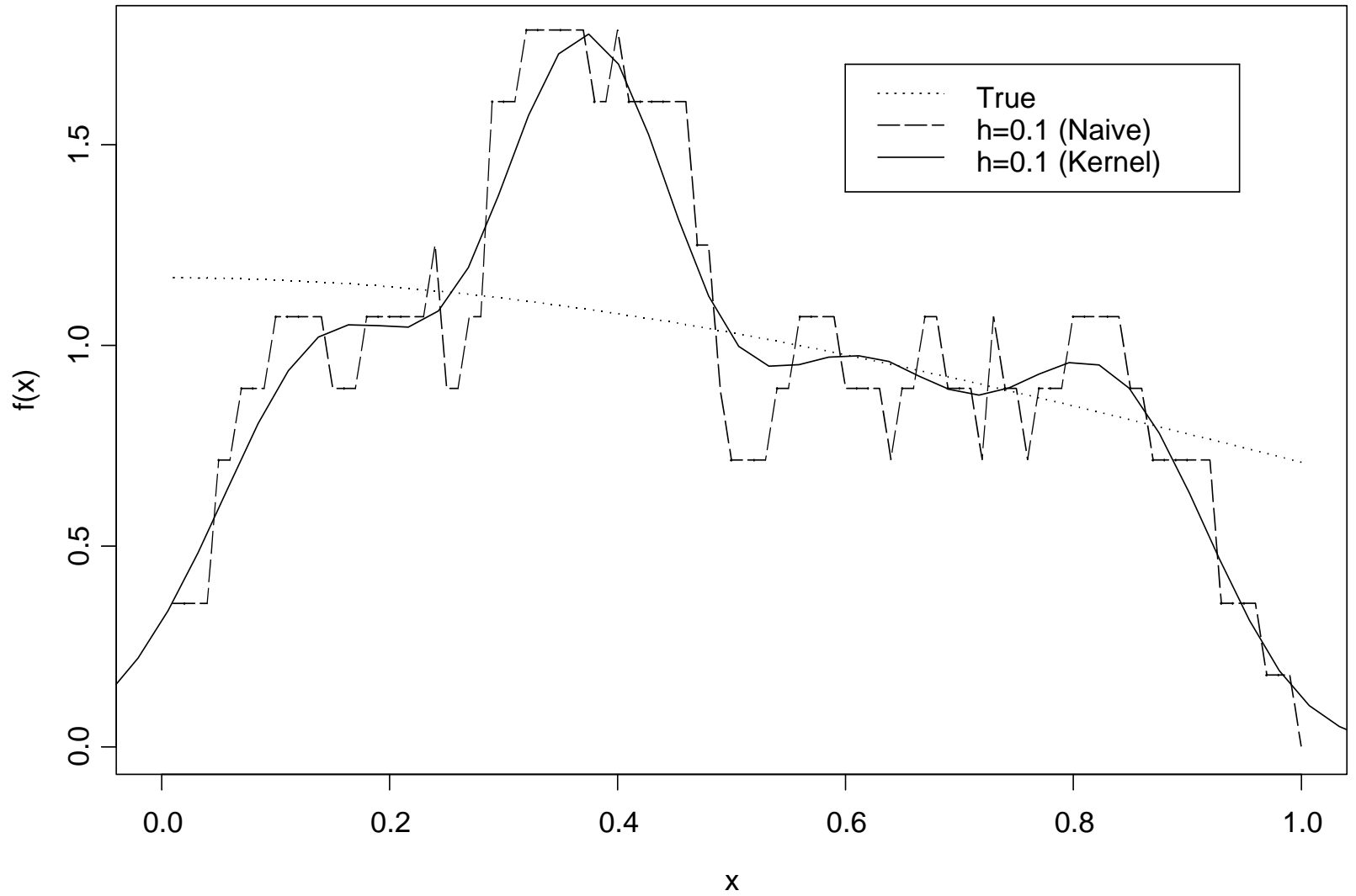
Guassian (i.e., normal), Cosine, Rectangular, Triangular, Lapalce.

Note: The choice of the bandwidth (i.e., h) is more critical than the kernel function.



Example of a Kernel Estimator

Nonparametric Density Estimates (n=28, N(0,1))

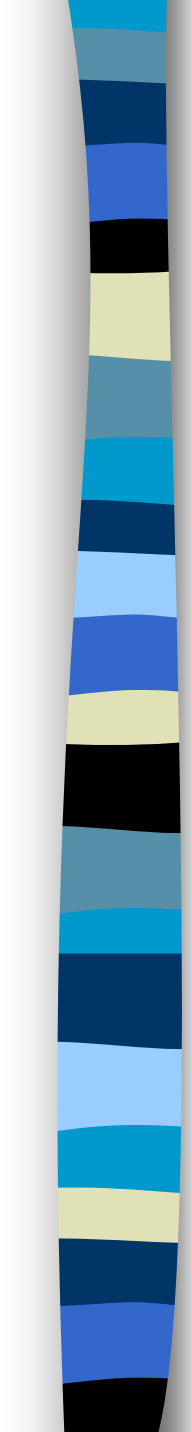




- Nearest-neighbor Estimator: (NNE)

→ Another usage of observations is to use the concept of “nearness” between points and observations. But, instead of distance, the nearness is measured according to the number of other observations between a point and the specified observation.

For example, if x_0 and x are adjacent in the ordering, then x is said to be 1-neighbor of x_0 . If another observation between x_0 and x , then x is said to be 2-neighbor of x_0 .



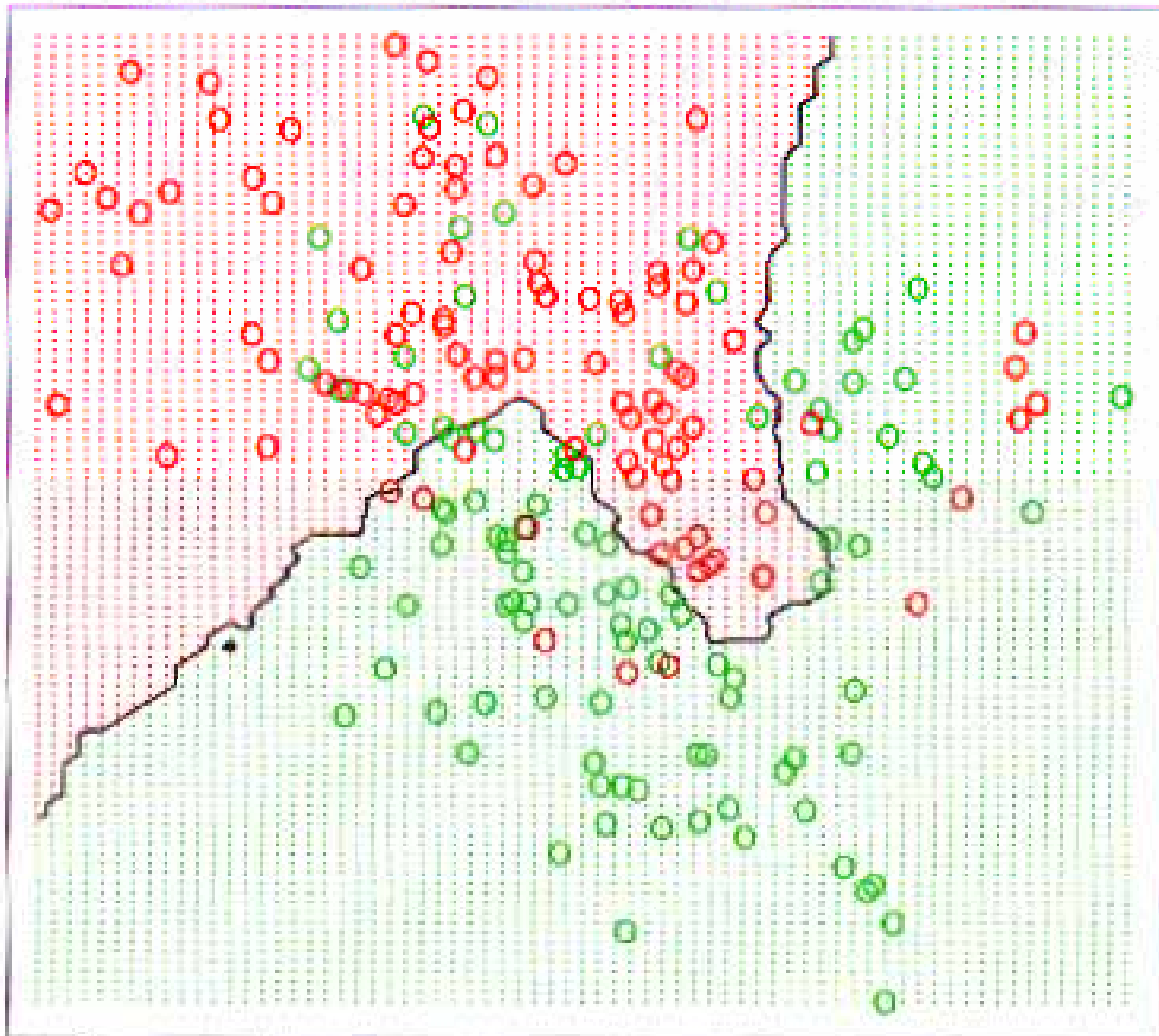
→ The nearest-neighbor density estimates are based on averages of the k nearest neighbors in the sample to the point x :

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_k(x)} K\left(\frac{x - x_i}{h_k(x)}\right),$$

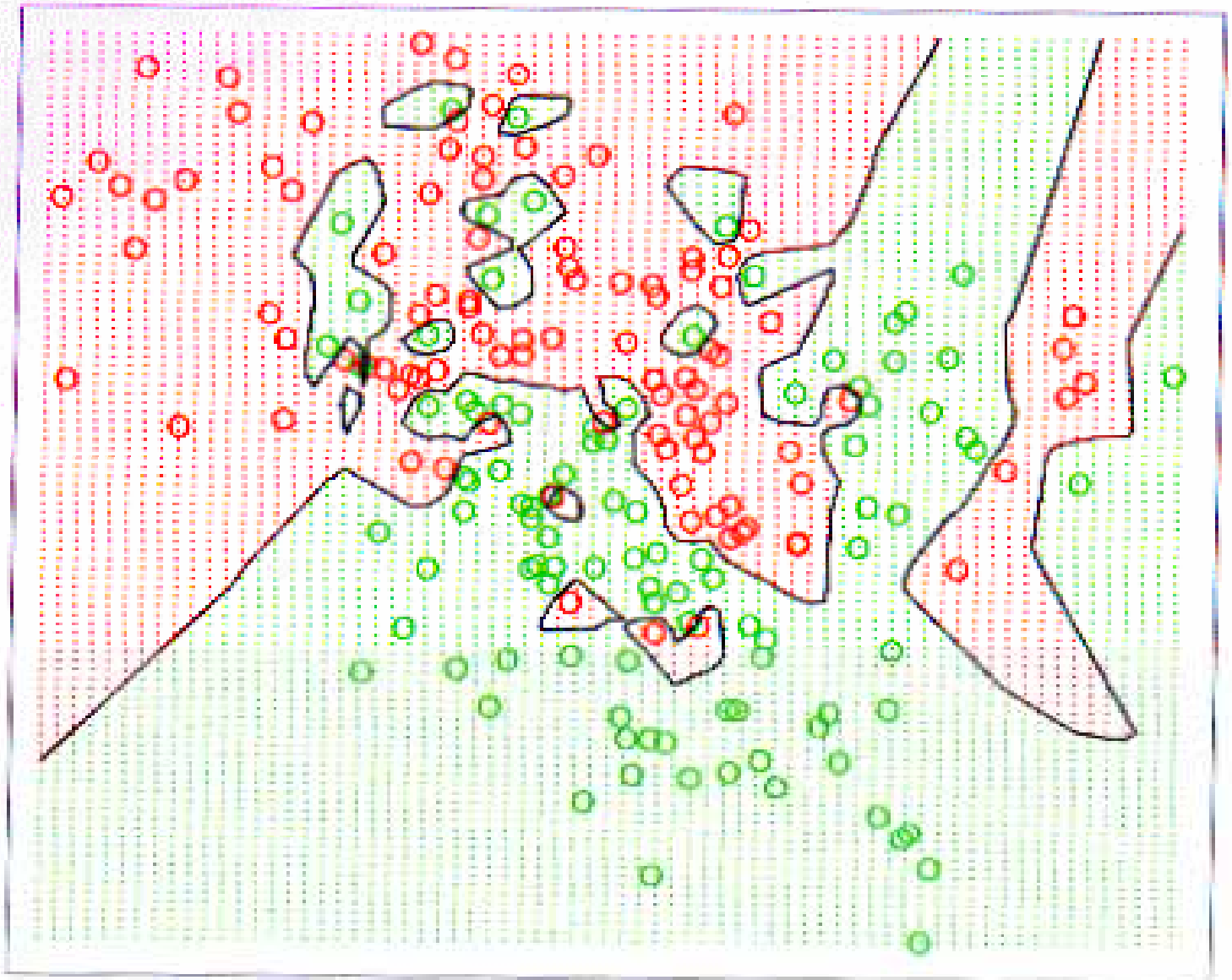
where $h_k(x)$ is the half-width of the smallest interval centered at x containing k data points.

Note: Unlike kernel estimates, the NNE use variable-width window.

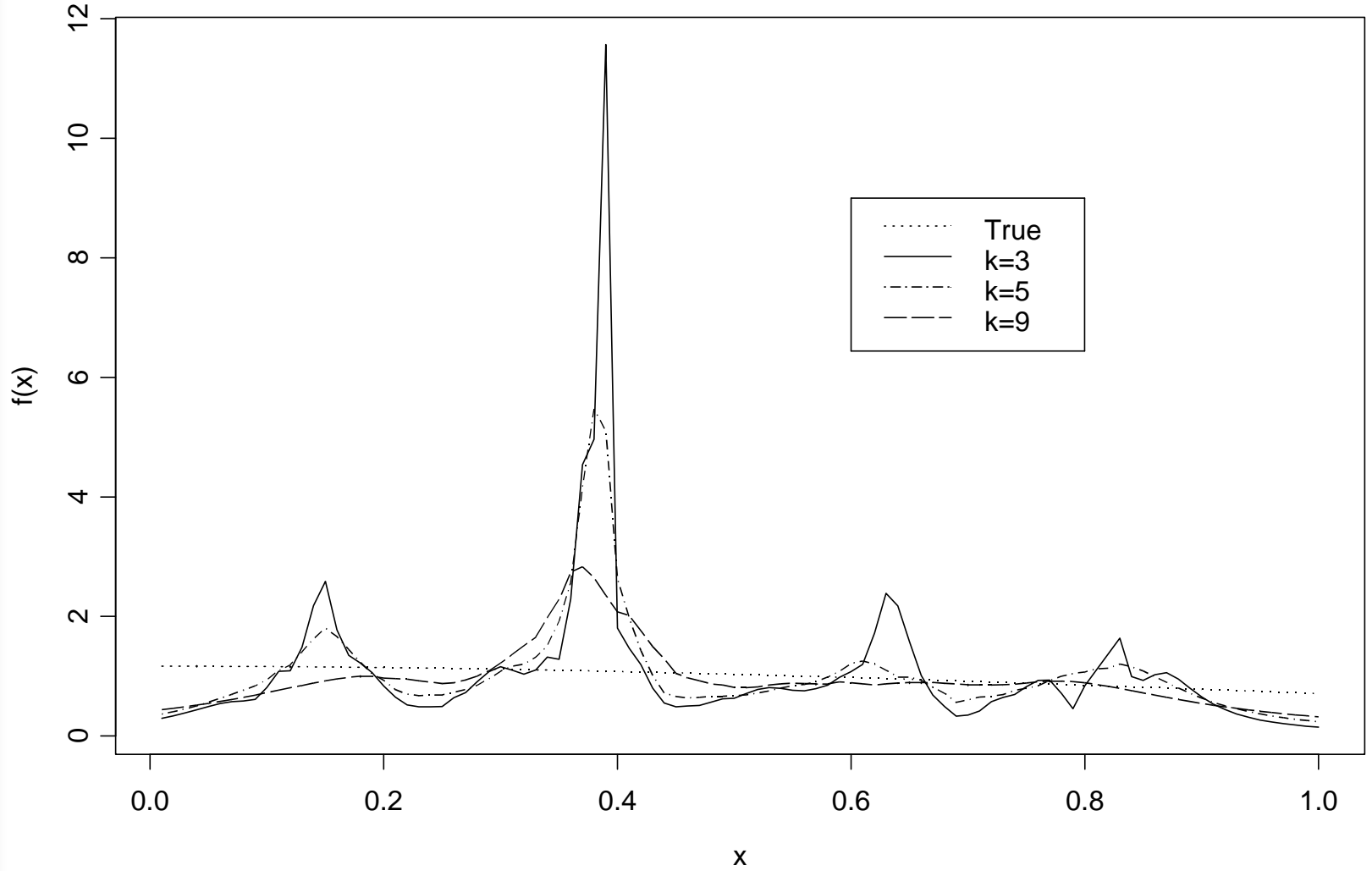
15-Nearest Neighbor Classifier



1-Nearest Neighbor Classifier



Nearest-neighbor Estimates (n=28, N(0,1))





■ Linear Smoother:

→ The goal is the smooth estimates \hat{M} of a regression function $M(x) = E(Y | X = x)$. A well-known example is the ordinary linear regression, where the fitted values are

$$\hat{y} = Hy, \text{ where } H = X(X'X)^{-1}X'.$$

→ A *Linear Smoother* is the one which the smooth estimate satisfies the following form:

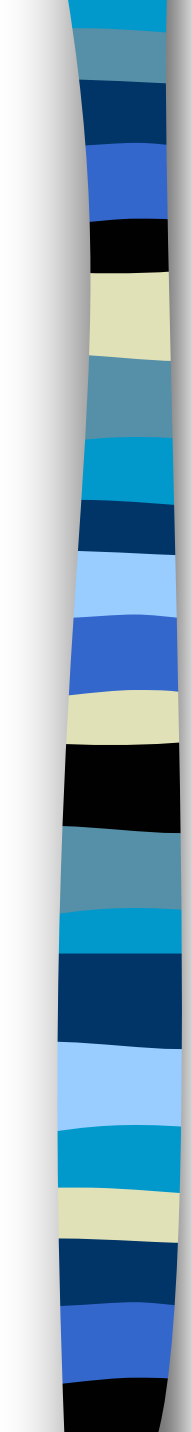
$$\hat{y} = S y,$$

where S is an $n \times n$ matrix depending on X .



■ Running Means:

- The simplest case is the running-mean smoother which computes \hat{y}_i by averaging y_j 's for which x_j falls in a neighborhood of x_i .
- One possible choice of the neighborhood N_i is to adapt the idea in *Nearest-neighbor* where N_i is the one with points x_j for which $|i-j| \leq k$. Such a neighborhood contains k points to the left and k points to the right. (Note: The two tails have fewer points and could be less smooth.)



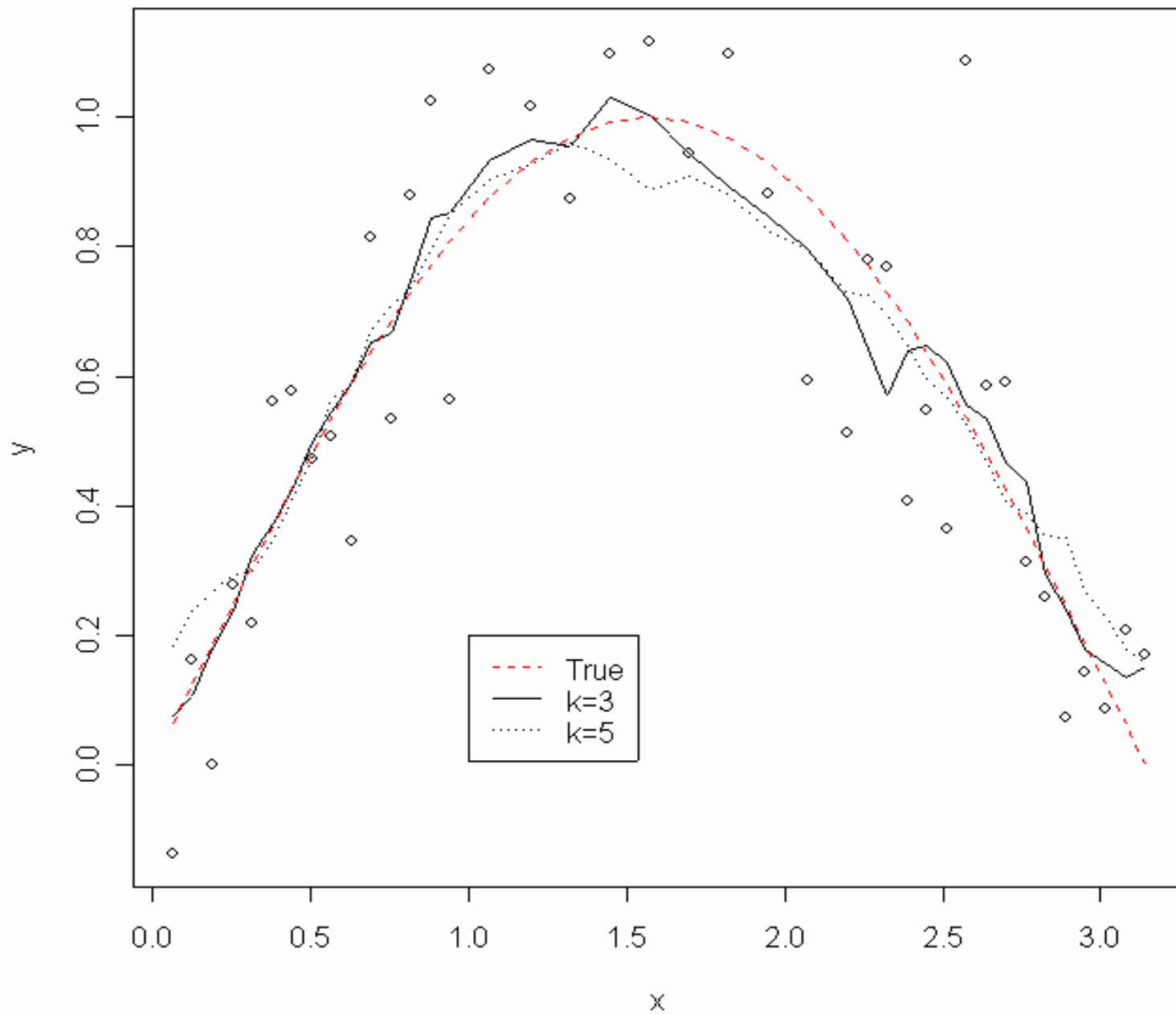
Note: The parameter k , called the *span* of the smoother, controls the degree of smoothing.

- Example 2. We will use the following data to demonstrate the linear smooth methods introduced in this handout. Suppose that

$$Y_i = \sin X_i + \varepsilon_i, \quad 0 \leq X_i \leq \pi,$$

where the noise ε_i is normally distributed with mean 0 and variance 0.04. Also, the setting of X is 15 points on $[0, 0.3\pi]$, 10 points on $[0.3\pi, 0.7\pi]$ and 15 points on $[0.7\pi, \pi]$.

Running means ($y=\sin x$)





■ Kernel Smoothers

→ The product of a running-mean smoother is usually quite unsmooth, since observations are getting equal weight regardless their distance to the point to be estimated. The kernel smoother with kernel K and window $2h$ uses

$$\hat{y}_i = \sum_{j \in N_i} w_{ij} y_j,$$

where

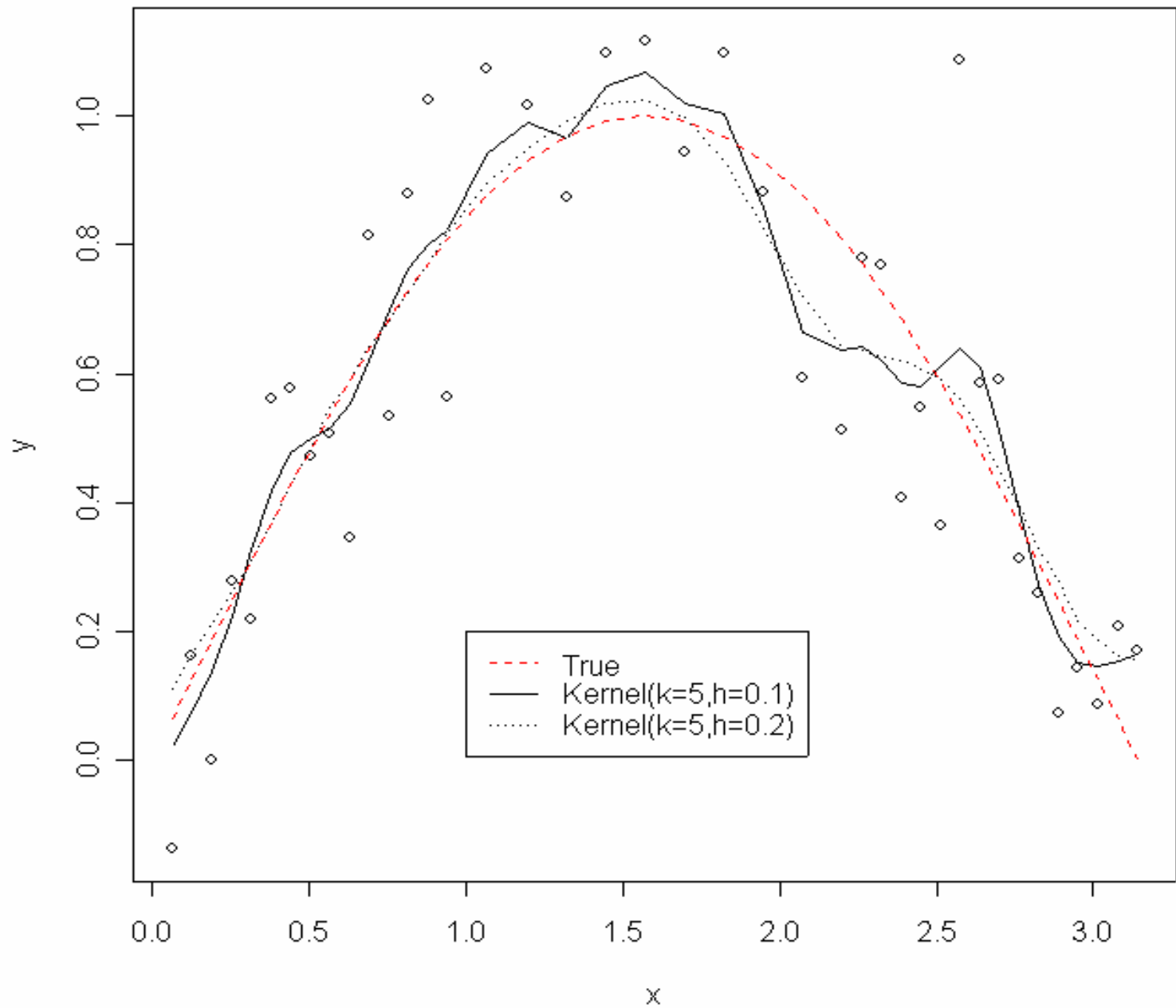
$$w_{ij} = K\left(\frac{x_i - x_j}{h}\right) / \sum_{j \in N_i} K\left(\frac{x_i - x_j}{h}\right)$$



Notes:

- (1) If the kernel is smooth, then the resulting output will also be smooth. The kernel smoother estimate can thus be treated as a weighted sum of the (smooth) kernels,
- (2) The kernel smoothers also cannot correct the problem of bias in the corners, unless the weight of observations can be negative.

Kernel Smoothers ($y=\sin x$)





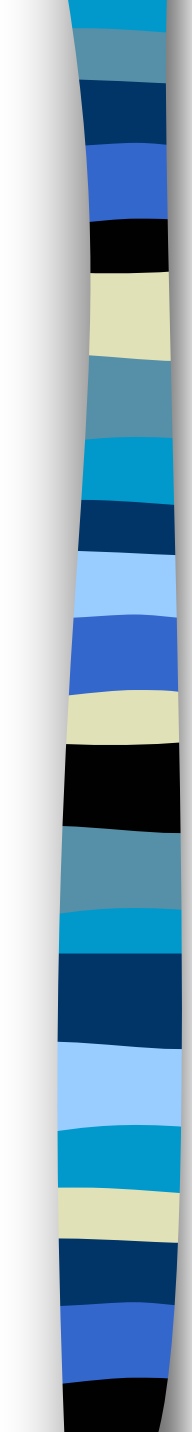
■ Spline Smoothing:

→ For the linear smoothers discussed previously, the smoothing matrix S is symmetric, has eigenvalues no greater than unity, and produce linear functions.

→ The smoothing spline is to select \hat{M} so as to minimize the following objective function:

$$S_{\lambda}(M) = \frac{1}{n} \sum_{i=1}^n [y_i - M(x_i)]^2 + \lambda \int_a^b [M''(t)]^2 dt,$$

where $\lambda \geq 0$ and $M \in C^3$.



Note: Two terms of the right-hand side of the objective function usually represent constraints opposite to each other.

→ The first term measures how far the smoothers differ from the original observations.

→ The second term, also known as *roughness penalty*, measures the smoothness of the smoothers.

Note: Methods which minimize the objective function are called *penalized LS methods*.



■ What are Splines?

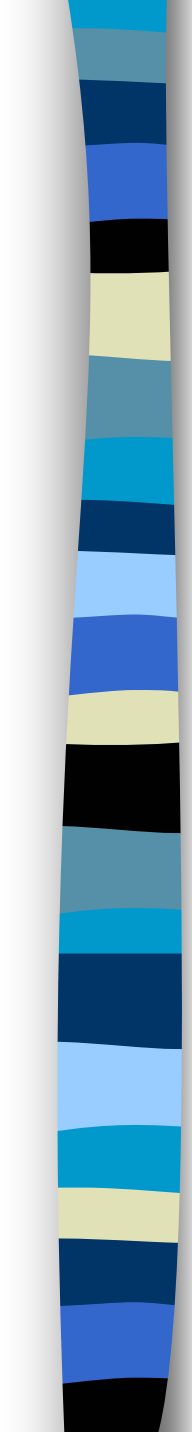
→ Spline functions, often called splines, are smooth approximating functions that behave very much like polynomials.

→ Splines can be used for two purposes:

(1) Approximate a given function (Interpolation)

(2) Smooth values of a function observed with noise

Note: We use terms “interpolating splines” and “smoothing splines” to distinguish.

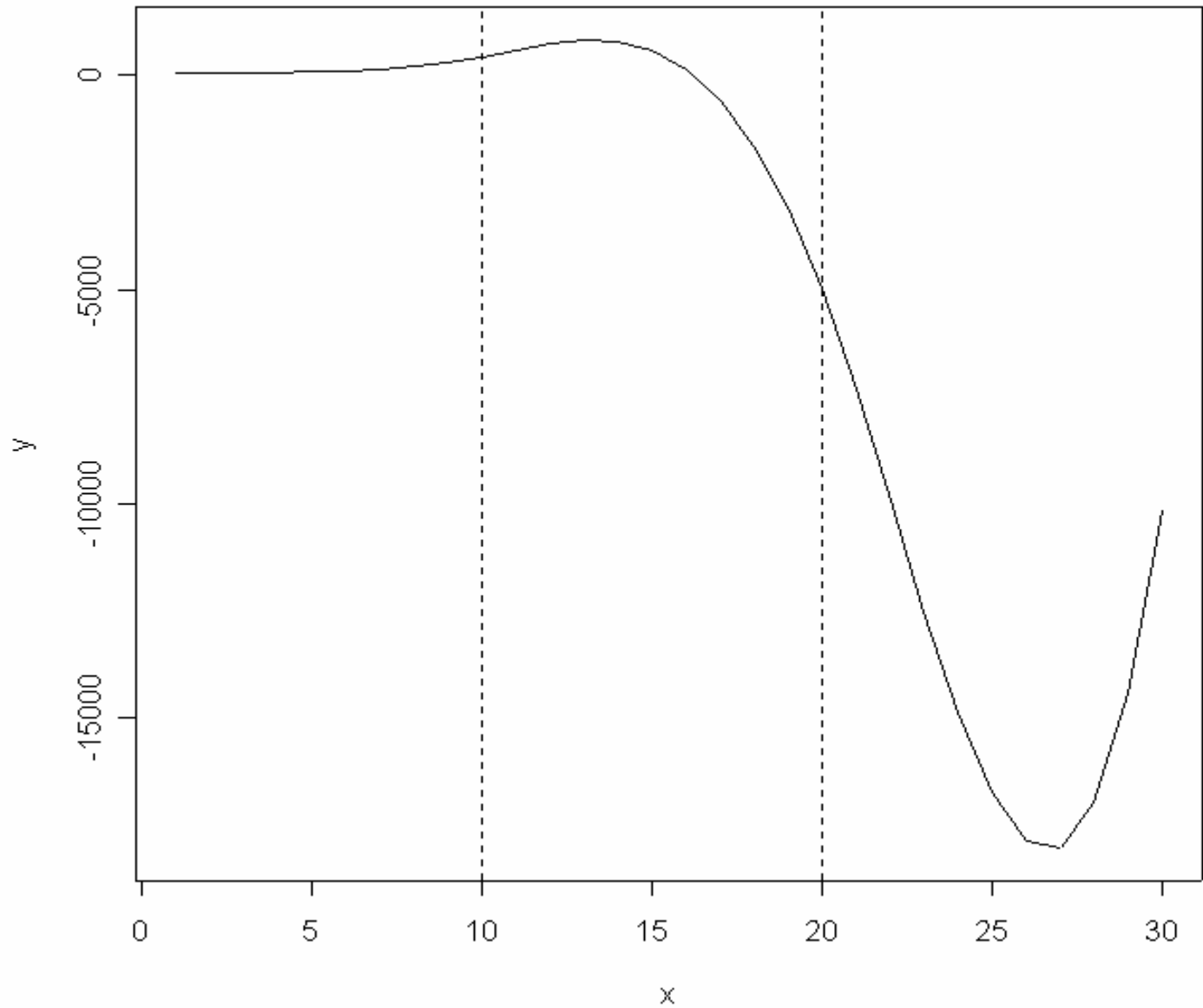


→ Loosely speaking, a spline is a piecewise polynomial function satisfying certain smoothness at the joint points. Consider a set of points, also named the set of knots, $K = \{x_1, x_2, \dots, x_m\}$ with $x_1 < x_2 < \dots < x_m$.

→ Piecewise-polynomial representations:

$$s(x) = \begin{cases} p_0(x) = p(x) & x < x_1 \\ p_1(x) & x_1 \leq x < x_2 \\ \dots & \dots \\ p_m(x) & x_{m-1} \leq x < x_m \\ p_{m+1}(x) & x_m \leq x \end{cases}$$

An Example of Cubic Splines



Q: Is it possible to use a polynomial to do the job?

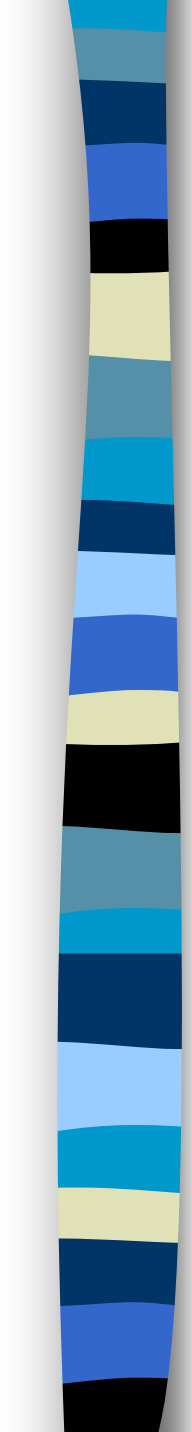
→ A cubic spline can be expressed as

$$P(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{i=4}^m \beta_i I(x \geq k_i)(x - k_i)^3,$$

which can also be expressed as $P = A\tilde{\beta}$,

where

$$A = \begin{bmatrix} 1 & 1 & 1^2 & 1^3 & \left. \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \right\} k_1 & \cdots \\ 1 & 2 & 2^2 & 2^3 & \vdots & \cdots \\ 1 & 3 & 3^2 & 3^3 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & 1^3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ 1 & n & n^2 & n^3 & (n - x_1)^3 & \cdots \end{bmatrix}, \quad \tilde{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \vdots \end{bmatrix}.$$



- Example 2. (continued)

→ We shall use cubic splines with knots at $\{0, \pi/3, 2\pi/3, \pi\}$ and compare the results of smoothing for different methods.

Note: There are also other smoothing methods available, such as LOWESS and running median (i.e., nonlinear smoothers), but we won't cover these topics in this class.

Linear Smoothers ($y=\sin x$)

