Introduction to Machine Learning

Chapter 8. Nonparametric Methods

Nov. 18, 2005, at ISS-AS

♠ Parametric: Bernoulli, multinomial, normal, MLE,etc. Multivariate methods: parameter estimation, classification, regression under normality.

♠ Semiparametric methods: mixture densities, clustering.

♠ **Nonparametric methods**

- density estimation
  - histogram
  - kernel estimator
  - k-nearest neighbor estimator
- regression
  - running mean smoother
  - kernel smoother
  - local polynomial fit, running line smoother
- classification

Parametric vs. nonparametric

Parametric: data are drawn from a probability distribution of **specific form** up to unknown parameters.

Semiparametric: in between, contains parametric and nonparametric components.

Nonparametric: data are drawn from a certain **unspecified** probability distribution.

Basic philosophy of nonparametric estimation/prediction

- The world is smooth and functions are changing slowly.
- Similar instances mean similar things.
- Unlike parametric methods, there is **no single global model**; **local models** are estimated as they are needed, affected only by closeby training data.
- Learn to know "similar patterns" from training set, and "interpolate" from them to find the right output (in prediction).
- Need a **distance measure** for similarity and interpolation.

  Different nonparametric algorithms differ in ways that they define similarity.

Heavier computational cost than parametric ones

In machine learning literature, nonparametric methods are also call **instance-based** or **memory-based learning** algorithms.

- Store the training instances in a lookup table and interpolate from these for prediction.

- **Lazy learning algorithm**, as opposed to the eager parametric methods, which have simple model and a small number of parameters, and once parameters are learned we no longer keep the training set.

Density estimation

Histogram

Training data: $\{x_i\}_{i=1}^n$ iid from a distribution with probability density function $p(x)$.

- Determine an origin and a bin width.

- Divide the space into equal sized bins with bin width h.

- $\widehat{p}(x) = \frac{\#\{x_i \text{ in the same bin as } x\}}{nh}$.

- Average shifted histogram: form histograms with different origins and average these histograms.

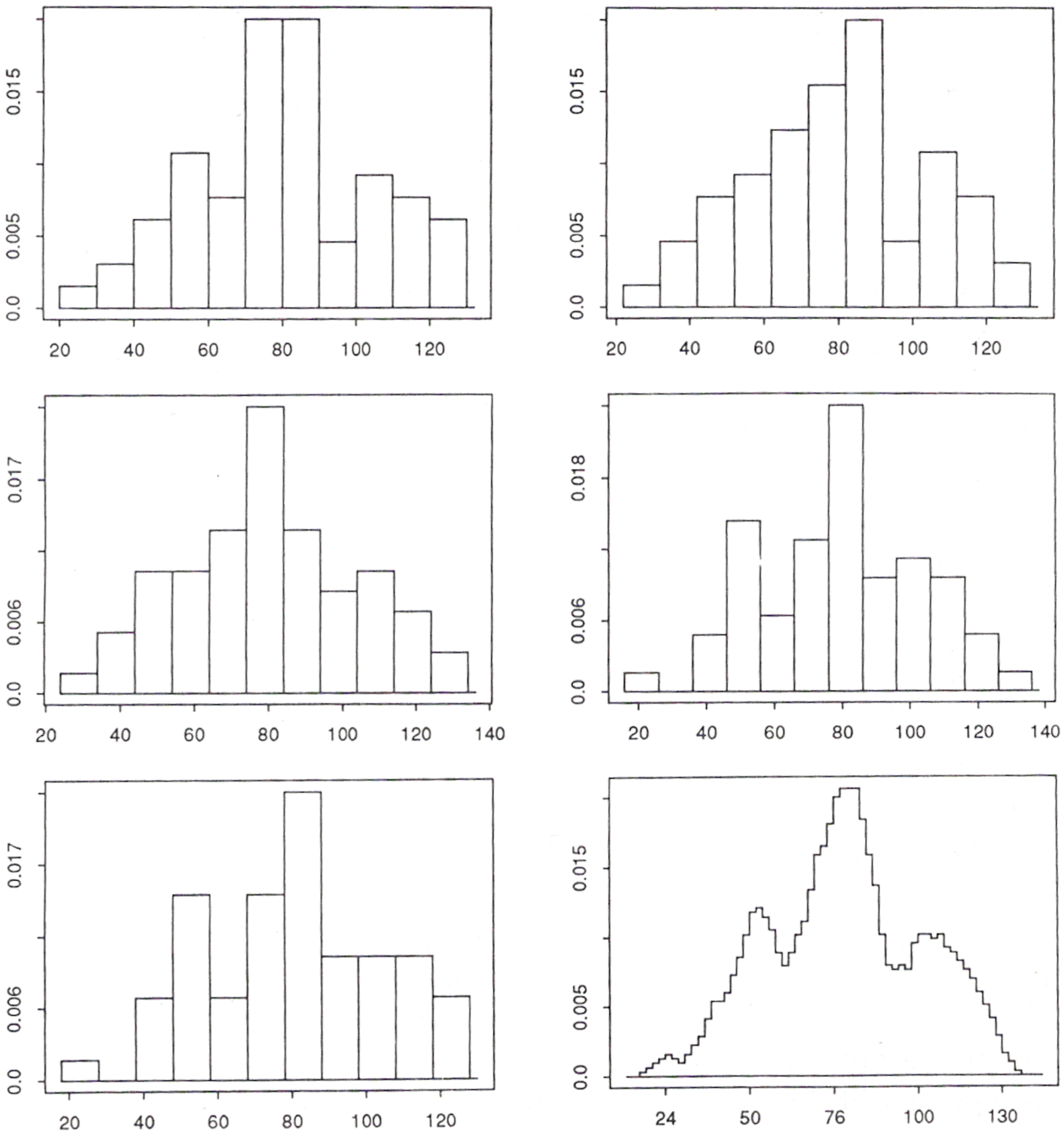


Figure 1.16 : Five histograms of the Buffalo snowfall data with the same binwidth $h = 10$, but with different origins $x_0 = 0, 2, 4, 6, 8$, and the average shifted histogram built from these five histograms.

Kernels as similarity measure

- Order 2 kernel $K(t)$: a pdf itself, $K(t) \geq 0$, $\int K(t)dt = 1$, $\int tK(t)dt = 0$, and $\int t^2K(t)dt > 0$.

- $K_h(t) = \frac{1}{h}K\left(\frac{t}{h}\right)$.

- similarity between x_1 and x_2: $K_h(x_1 - x_2)$.

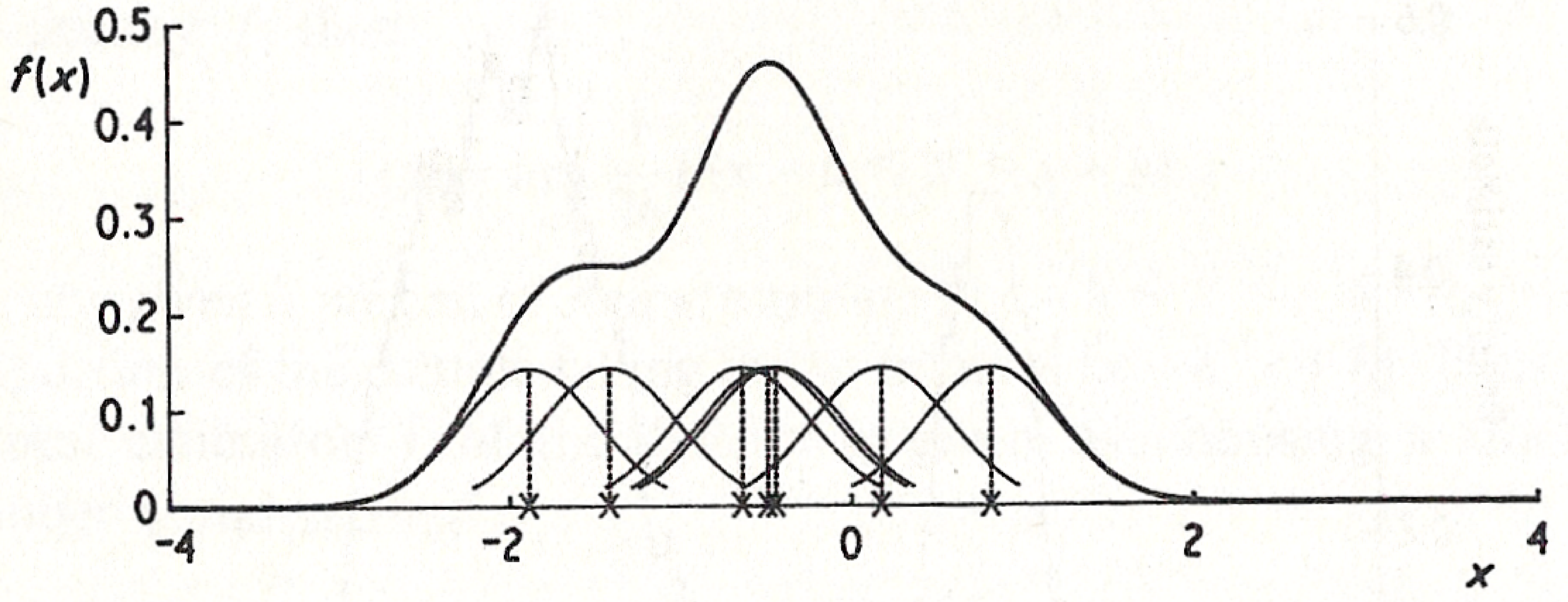


Fig. 2.4 *Kernel estimate showing individual kernels. Window width 0.4.*

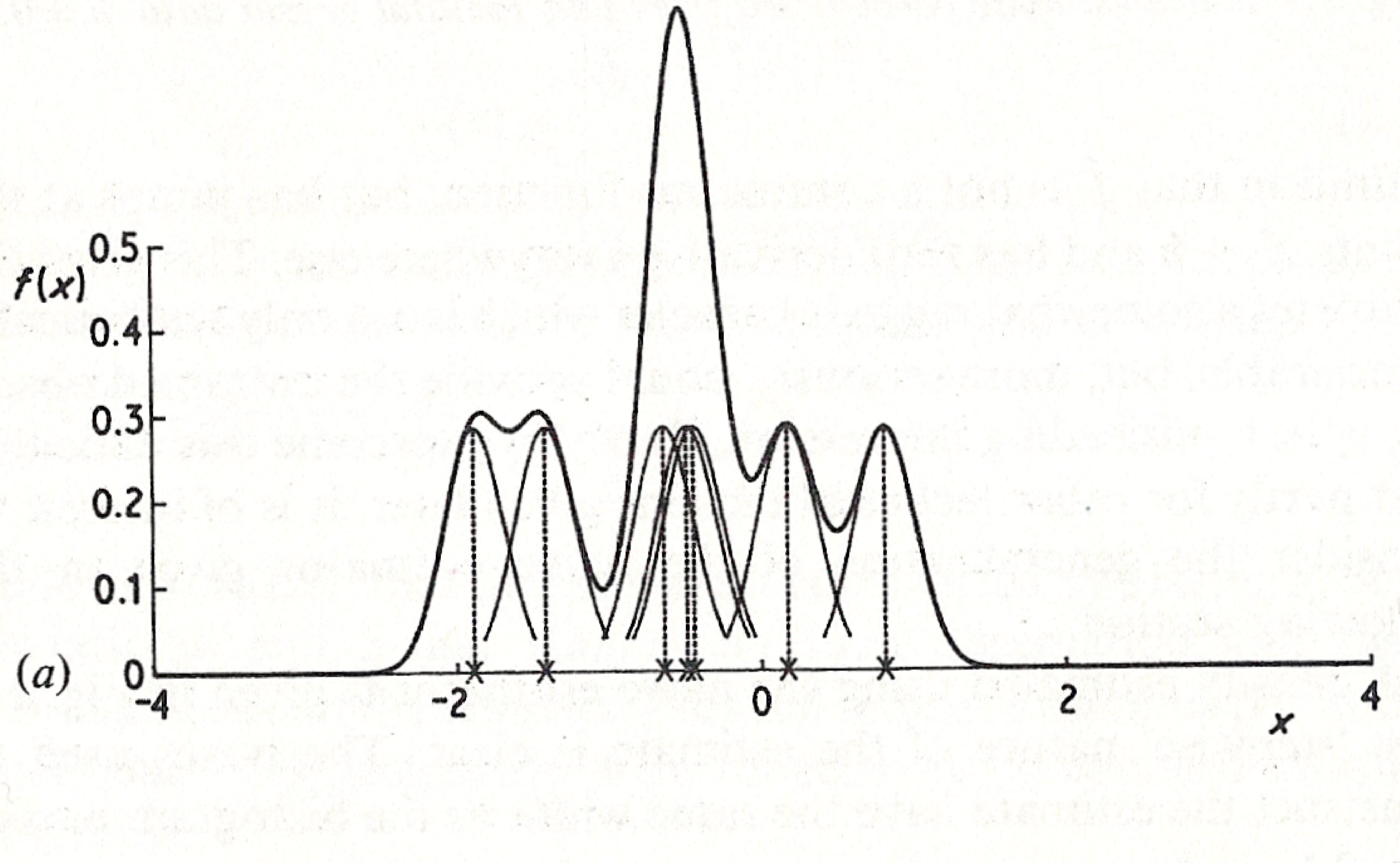


Fig. 2.5 *Kernel estimates showing individual kernels. Window widths: (a) 0.2; (b) 0.8.*

Kernel estimator

- Choose a kernel as weight function.

- Decide a window width.

- $\widehat{p}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)$.

- Small h: each training instance has a large effect in a small region and no effect on distant points.

  Larger h: weight function is flatter and more spread out. There is more overlap of the kernels and we get a smoother estimate.

- Ideally: use a varying adaptive window width; smaller h for dense-data region and larger h for sparse-data region.

k-nearest neighbor estimator

- It adapts the amount of smoothing to the local density of data.

- The probability that a point x falls within V centered at x: $\theta = \int_V p(t)dt \approx p(x)V \approx k/n$.

  naive k-nearest neighbor estimator: $\widehat{p}(x) = \frac{k}{nV}$, $V = 2d_k(x)$.

- The degree of smoothing is controlled by k, the number of neighbors taken into account.

- $\widehat{p}(x) = \frac{1}{nd_k(x)} \sum_{i=1}^{n} K\left(\frac{x-x_i}{d_k(x)}\right)$, kernel k-nearest neighbor.

  This is a kernel estimator with adaptive variable window width.

Generalization to multivariate data

- product kernel: $K(\mathbf{t}) = \prod_{j=1}^{d} K(t_j)$.

- d-dimensional observations, the multivariate kernel density estimator: $\widehat{p}(x) = \frac{1}{n\mathbf{h}^{\mathbf{d}}} \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)$.

- **Curse of dimensionality**. Think of 8-dimensional histogram with 10 bins per dimension, then there are 10^8 bins in total. Unless we have enormous amount of data, most of these bins will be empty.

- **Instability, high variation in estimation/prediction.**

- naive k-nn estimator: $\widehat{p}(x) = \frac{k}{nV}$, V: volume of d-dimensional ball with radius $d_k(x) = \|x - x_{(k)}\|$.

Sample size required (accurate to about 3 significant figures) to ensure that the relative mean square error at zero is less than 0.1, when estimating a standard multivariate normal density using a normal kernel and the window width that minimizes the mean square error at zero.

Dimension	Required sample size
1	4
2	19
3	67
4	223
5	768
6	2790
7	10700
8	43700
9	187000
10	482000

the relative mean square error at zero: $E(\hat{f}(0) - f(0))^2/f^2(0)$.

Things you must learn from this course **Dimension reduction**

- Dimension reduction: subset (variables) selection, PCA, factor analysis, multi-dimensional scaling, linear discriminant analysis, SIR, etc.

- CCA (canonical correlation analysis).

- Most methods are based on **spectral analysis**.

  Eigen-decomposition, elicit leading eigen-components, or

  Singular value decomposition.

  SIR: Eigen-decomposition of between group (slice) covariance with respect to Σ_X.

  Linear discriminant analysis.

- Support vector machines sequel: SVM classification, SVR, reduced SVM, etc.

Singular value decomposition

$$[\ \mathbb{X},\ \ \mathbb{Y}\ ] = \begin{bmatrix} x_1' & y_1' \\ \vdots & \vdots \\ x_n' & y_n' \end{bmatrix}_{n\times(p+q)} .$$

e.g., Cov$(X) = \mathbb{X}'\mathbb{X}$ and Cov$(X,Y) = \mathbb{X}'\mathbb{Y}$. (assume centered)

SVD: $\mathbb{X}'\mathbb{Y} = \mathbb{U}_{p\times p}\ \mathbb{D}_{p\times q}\ \mathbb{V}'_{q\times q}$, where $\mathbb{U}, \mathbb{V}$ orthogonal, $\mathbb{D}$ diagonal.

$(\mathbb{X}\mathbb{U})'\ (\mathbb{Y}\mathbb{V}) = \mathbb{D}$.

$\mathbb{U}$ and $\mathbb{V}$: two new coordinate systems for R^p and R^q respectively.

Nonparametrics + Dimension reduction

concept first, then technique.

Regression

Parametric vs. nonparametric: global vs. local models

- Given the iid training data $\{(x_i, y_i)\}_{i=1}^{n}$, where $y_i = g(x_i) + \epsilon_i$. Assume ϵ_i, x_i independent, $E\epsilon_i = 0$, $Var(\epsilon_i) = \sigma^2$.

- $g(x)$: regression surface;

  parametric: e.g., regression line; a global model;

  nonparametric: e.g., mixture of kernels, local polynomials.

- y: regression surface observed with noise.

Regresssorgram

- $\widehat{g}(x) = \sum_{i=1}^{n} w_i(x) y_i$ with $\sum_{i=1}^{n} w_i(x) = 1$. Or equivalently,

  $\widehat{g}(x) = \sum_{i=1}^{n} w_i(x) y_i / \sum_{i=1}^{n} w_i(x)$.

- Partition the interval (or region) into bins.

- $w_i(x) = \begin{cases} 1 & \text{if } x_i \text{ is in the same bin with } x \\ 0 & \text{otherwise} \end{cases}$

running mean smoother

- $\widehat{g}(x) = \frac{\sum_{i=1}^{n} w_h(x-x_i) y_i}{\sum_{i=1}^{n} w_h(x-x_i)}$, $\quad w_h(t) = \frac{1}{h}$ if $|t| \leq h$, zero otherwise.

Kernel estimator, kernel smoother

- $\widehat{g}(x) = \frac{\sum_{i=1}^{n} w_h(x-x_i) y_i}{\sum_{i=1}^{n} w_h(x-x_i)}$, $\quad w_h(t) = \frac{1}{h}$ if $|t| \leq h$, zero otherwise.

  Uniform kernel weight function.

- Replace the above weight function (which is a uniform kernel) by a general kernel K.

- $\widehat{g}(x) = \frac{\sum_{i=1}^{n} K_h(x-x_i) y_i}{\sum_{i=1}^{n} K_h(x-x_i)}$.

- k-nearest neighbor smoother: take $h = d_k(x)$.

Local polynomials regression –local constant fit

Parametric: global model; bias and variance issues.
Nonparametric: local model; bias, variance.

- fitting criterion: in a small region around x_0, $g(x) \approx a_0$,

$$\hat{a}_0 = \arg\min_{a_0} \sum_{i=1}^{n} (y_i - a_0)^2 w_i, \quad \sum_i w_i = 1.$$

- Take derivative wrt a_0, set it to zero. $\hat{a}_0 = \sum_{i=1}^n y_i w_i$.

- Kernel weights: $w_i = K_h(x_0 - x_i) / \sum_{i=1}^n K_h(x_0 - x_i)$.

$\hat{g}(x_0) = \frac{n^{-1} \sum_{i=1}^n y_i K_h(x_0 - x_i)}{n^{-1} \sum_{i=1}^n K_h(x_0 - x_i)}$.

$\hat{g}(x) = \frac{n^{-1} \sum_{i=1}^n y_i K_h(x - x_i)}{n^{-1} \sum_{i=1}^n K_h(x - x_i)}$: Nadaraya-Watson kernel est.

Local polynomials regression –local linear fit

- fitting criterion: in a small region around x_0,
  $g(x) \approx a_0 + b_0(x - x_0)$,
  $$\hat{a}_0 = \arg\min_{a_0}\min_{b_0}\sum_{i=1}^{n}(y_i - a_0 - b_0(x_i - x_0))^2 w_i, \quad \sum_i w_i = 1.$$

- Kernel weights: $w_i = K_h(x_0 - x_i)/\sum_{i=1}^{n} K_h(x_0 - x_i)$.

- Homework-IV, problem-1: $\hat{a}_0 =?$ $\quad \hat{g}(x) =?$

Homework problem 1, due 11/25

Assume we have iid data $\{(x_i, y_i)\}_{i=1}^n$, where $y_i = g(x_i) + \epsilon_i$. Suppose that $g(x)$ is approximated locally by a linear polynomial with kernel weight function $K_h(x - x_i)$.

- fitting criterion: in a small region around x_0, $g(x) \approx a_0 + b_0(x - x_0)$,

$$(\widehat{a}_0, \widehat{b}_0) = \arg\min_{a_0, b_0} \sum_{i=1}^{n} (y_i - a_0 - b_0(x_i - x_0))^2 w_i$$

- Kernel weights: $w_i = K_h(x_0 - x_i) / \sum_{i=1}^n K_h(x_0 - x_i)$.

Derive the local linear estimator $\widehat{g}(x)$.

Running line smoother (LOWESS)

locally weighted scatter plot smoothing

- Fit a local linear polynomial via the method on the last slide.
- Calculate residuals, $r_k = y_k - \widehat{y}_k$, and assign weight to each residual, $\delta_k = B(r_k/\text{median}(|r_1|, \ldots, |r_n|)$, where $B(t) = (1 - |t|^2)^2$. New weights for observations: $w_i^{\text{new}}(x) = \delta_i w_i^{\text{orig}}(x)$.
- Carry through again a local linear fit with new weights. Observations showing large residuals in the initial fit are **downweighted** in the second fit.
- Repeat a number of times.

Purpose: to robustify against outliers and to further smooth the local polynomial fit.

Choice of smoothing parameter

In nonparametric methods, for density estimation or regression, **one of the critical things is the smoothing parameter**.

- Histogram bin width h.
- Kernel window width h.
- The number of neighbors k in nearest-neighbor estimator.
- Small h or k leads to small bias but large variance. Larger h or k decreases variance but increases bias.

Choice of smoothing parameter -cross validation

- **Leave-one-out** cross-validation: use $n-1$ sample data for training and test on the remaining one. This is repeated for all n subset of size $n-1$. Computationally expensive.

- ν**-fold** cross-validation: partition the training set into ν subsets, train on $\nu-1$ subsets and test on the remaining one. This procedure is repeated as each subset is withheld in turn.

Classification

Nonparametric classification via class-conditional densities -kernel approach

- Class conditional densities: $p(x|C_i)$.

- $\widehat{p}(x|C_i) = \frac{1}{n_i h^d} \sum_{j=1}^{n_i} K(\frac{x-x_j}{h})$, x_j from class C_i.

- Estimates for class distribution: $\widehat{P}(C_i) = n_i/n$,

  n_i: no. of data from C_i, n: total no. of data.

- Discriminant rule: assign x to the class which takes the maximum among $\widehat{p}(x|C_i)\widehat{P}(C_i)$.

  $x \to \arg\max_i \ \widehat{p}(x|C_i)\widehat{P}(C_i)$.

Nonparametric classification via class-conditional densities -k nearest neighbor approach

- $\widehat{p}(x|C_i)P(C_i) = \frac{k_i}{n_i V(x)} \cdot \frac{n_i}{n} \propto k_i$.

  Assign x to the class having most examples among the k-neighbors of the input. All neighbors have equal vote, and the class having the maximum number of voters among the the k neighbors is chosen.

- k_i: no. of neighbors out of the k nearest that belong to C_i.

- $V(x)$: the volume of a d-dimensional ball with radius $d_k(x)$.

Kernels

Examples of kernels

Kernel	$K(u)$
Uniform	$\frac{1}{2}I(\lvert u\rvert \leq 1)$
Triangle	$(1-\lvert u\rvert)I(\lvert u\rvert \leq 1)$
Epanechnikov	$\frac{3}{4}(1-u^2)I(\lvert u\rvert \leq 1)$
Quartic	$\frac{15}{16}(1-u^2)^2I(\lvert u\rvert \leq 1)$
Triweight	$\frac{35}{32}(1-u^2)^3I(\lvert u\rvert \leq 1)$
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}u^2)$
Cosinus	$\frac{\pi}{4}\cos(\frac{\pi}{2}u)I(\lvert u\rvert \leq 1)$

Approximation by kernel convolution

- For $p, K \in L_1(R)$, we define their convolution $p * K$ as

$$(p * K)(x) = \int p(x-t)K(t)dt = \int K(x-t)p(t)dt.$$

- For $p(x)$ being a pdf, $(p * K)(x) = \int K(x-t)dP(t)$, a natural empirical estimate is $\widehat{p}(x) = n^{-1}\sum_{i=1}^{n} K_h(x-x_i)$.

- Systematic bias is caused by convolution approximation.

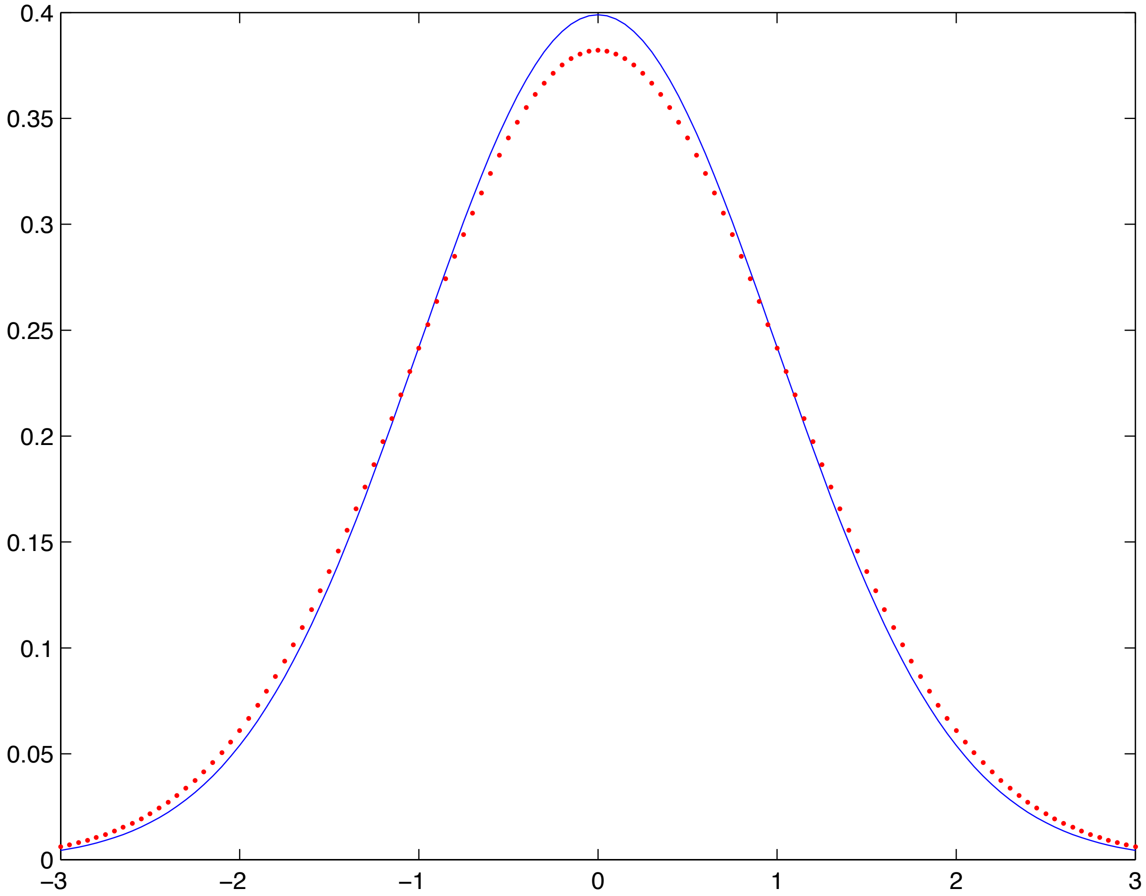
- For $g(x)$ being a regression function,

$$(K * g)(x) = \int K(x-t)g(t)dt = \int \frac{K(x-t)g(t)}{p(t)}dP(t),$$

  a natural empirical estimate is

  $\widehat{g}(x) = n^{-1}\sum_{i=1}^{n} K_h(x-x_i)y_i/\widehat{p}(x).$

Kernel convolution



True: blue curve, convolution approximation: red dotted curve