Non-parametric methods

kNN classifier, Kernel density estimate, Kernel regression

Aarti Singh

Co-instructor: Pradeep Ravikumar

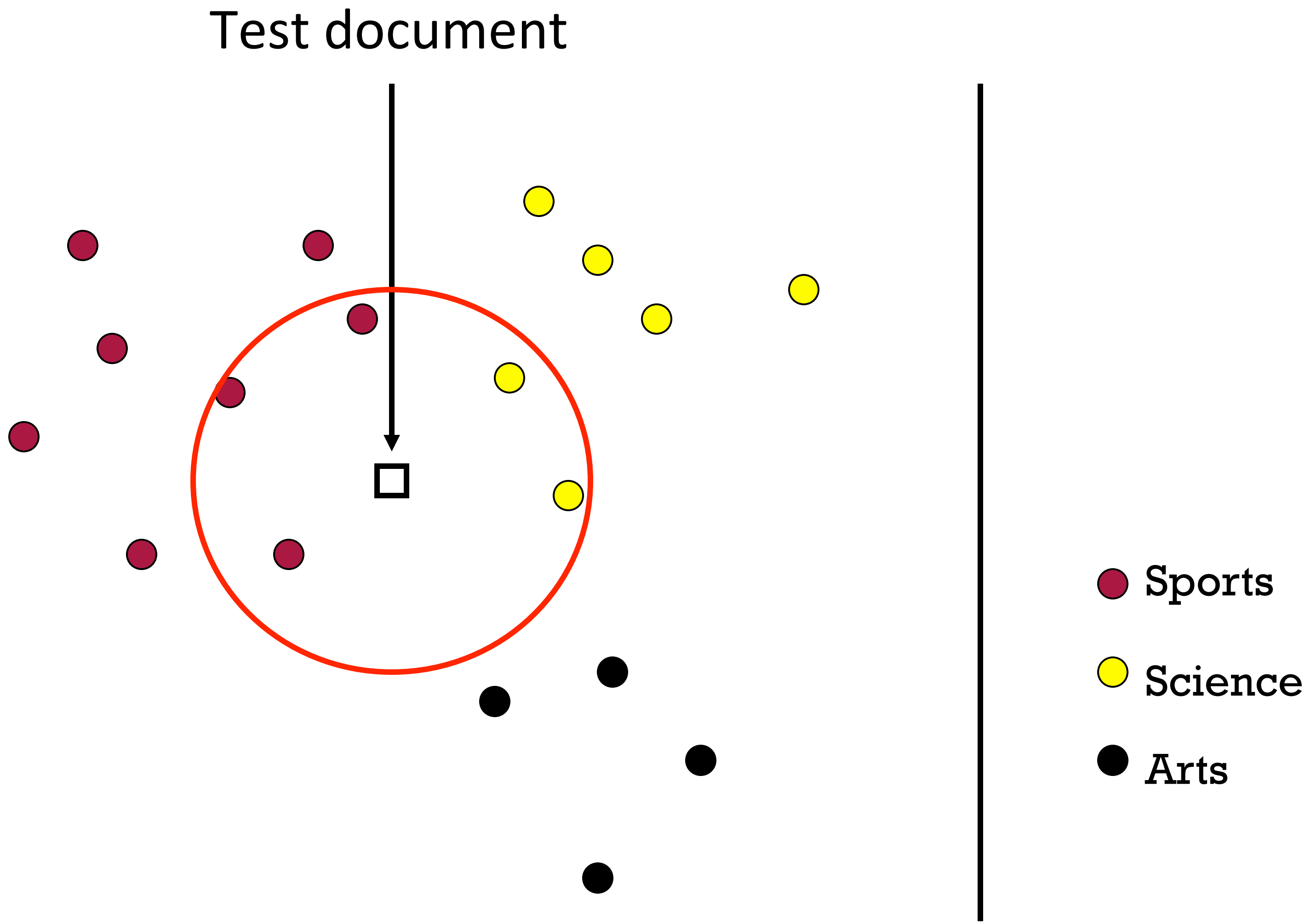
Machine Learning 10-701
Mar 6, 2017

Non-Parametric methods

- Typically don't make any distributional assumptions
- As we have more data, we should be able to learn more complex models
- Let number of parameters scale with number of training data

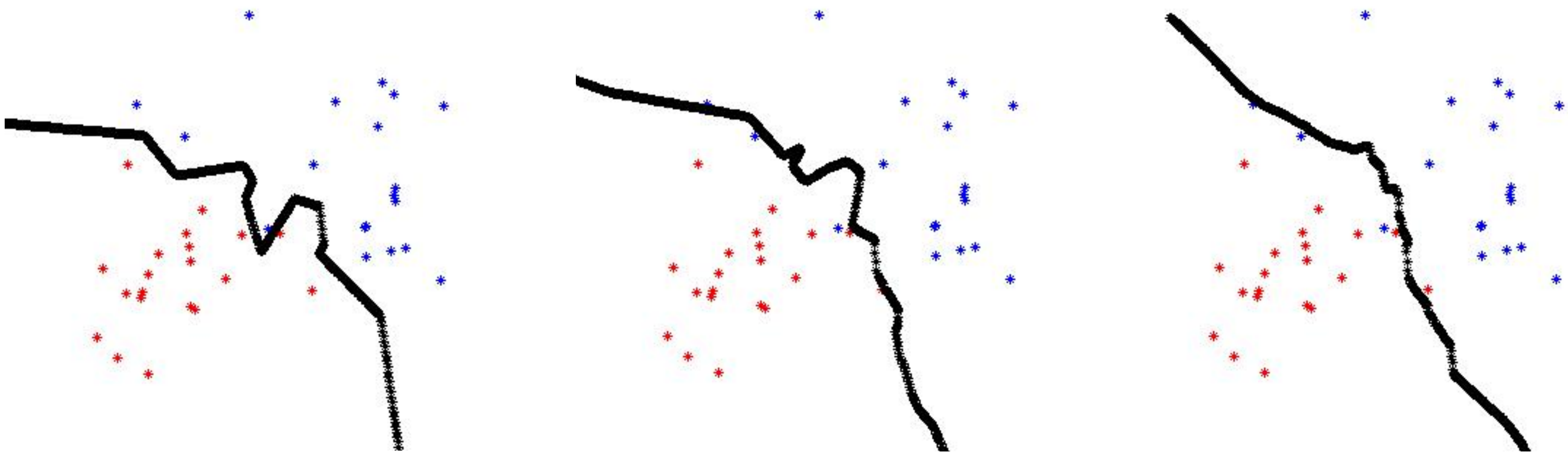
- We will see some nonparametric methods for
  - Classification
  - Density estimation
  - Regression

k-NN classifier (k=5)



What should we predict? … Average? Majority? Why?

k-NN classifier – decision boundary



1NN 5NN 9NN

- K acts as a smoother (Bias-variance tradeoff)

Case Study:
kNN for Web Classification

- Dataset
  - 20 News Groups (20 classes)
  - Download :(http://people.csail.mit.edu/jrennie/20Newsgroups/)
  - 61,118 words, 18,774 documents
  - Class labels descriptions

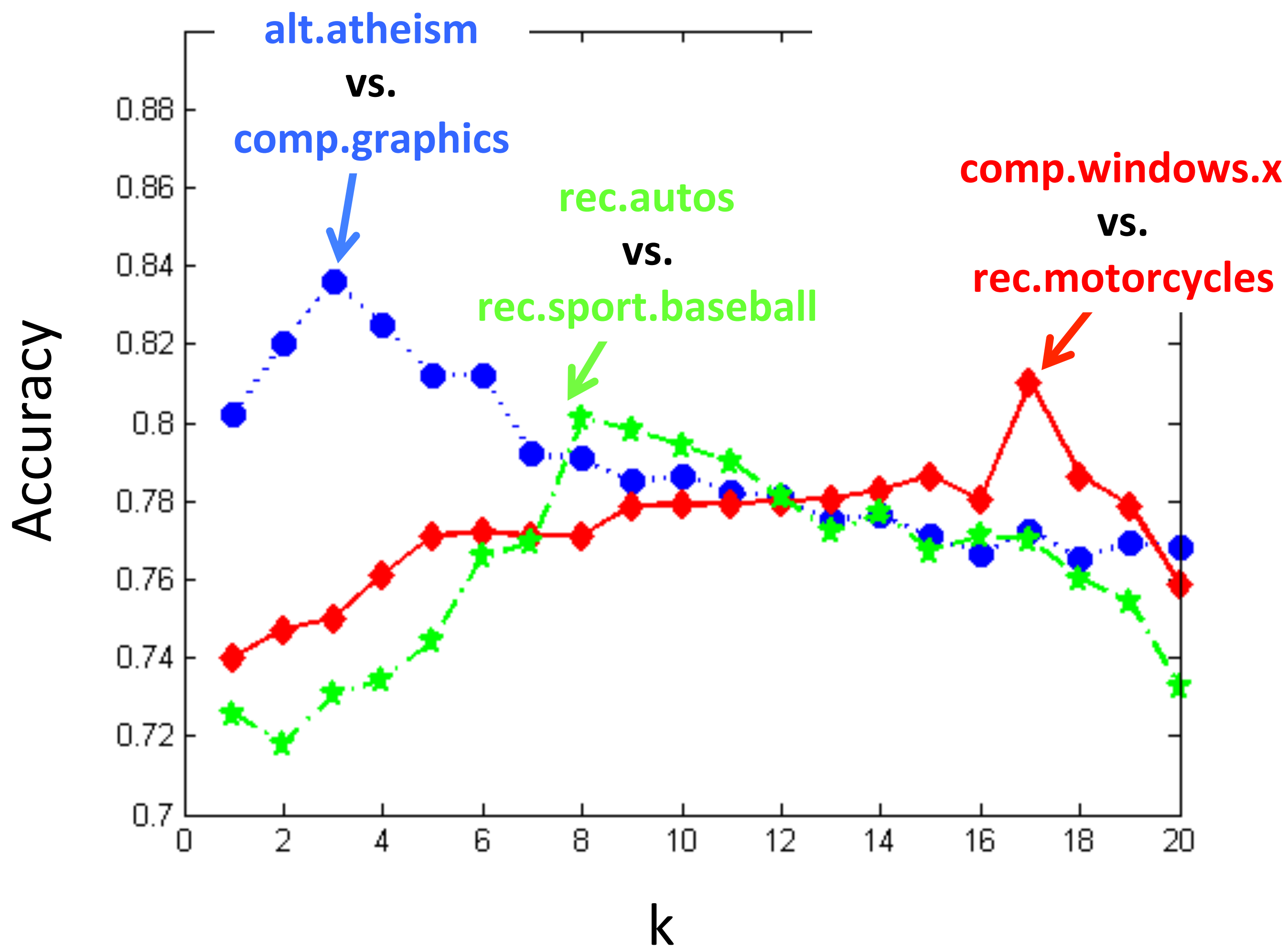
comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

Experimental Setup

- Training/Test Sets:
  - 50%-50% randomly split.
  - 10 runs
  - report average results
- Evaluation Criteria:

$$Accuracy = \frac{\sum_{i \in test\ set} \mathrm{I}(predict_i = true\ label_i)}{\#\ of\ test\ samples}$$

Results: Binary Classes



k-NN classifier

- Optimal Classifier:
$$\begin{aligned} f^*(x) &= \arg\max_y P(y|x) \\ &= \arg\max_y P(x|y)P(y) \end{aligned}$$

- k-NN Classifier:
$$\begin{aligned} \widehat{f}_{kNN}(x) &= \arg\max_y \ \widehat{P}_{kNN}(x|y)\widehat{P}(y) \\ &= \arg\max_y \ k_y \end{aligned}$$

Lets consider discrete features first:

$$\widehat{P}_{kNN}(x|y) = \frac{k_y}{n_y}$$

$k_y \rightarrow$ # training pts of class y that have feature value(s) x

$n_y \rightarrow$ # total training pts of class y

$$\widehat{P}(y) = \frac{n_y}{n}$$

$n_y \rightarrow$ # training pts of class y

$n \rightarrow$ # total training pts

k-NN classifier

- Optimal Classifier: $$f^*(x) = \arg\max_y P(y|x) = \arg\max_y P(x|y)P(y)$$

- k-NN Classifier: $$\widehat{f}_{kNN}(x) = \arg\max_y \ \widehat{P}_{kNN}(x|y)\widehat{P}(y) = \arg\max_y \ k_y$$

Lets consider discrete features first:

$$\widehat{P}_{kNN}(x|y) = \frac{k_y}{n_y}$$

k_y → # training pts of class y that have feature value(s) x

n_y → # total training pts of class y

What if no training pts of class y have feature values x? Almost surely the case with continuous features.

k-NN classifier

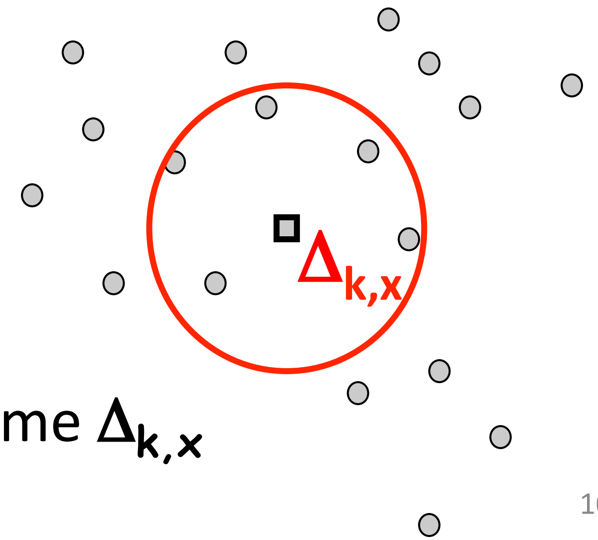
- Optimal Classifier: $f^*(x) = \arg\max_y P(y|x)$
$$= \arg\max_y p(x|y)P(y)$$

Prob density

- k-NN Classifier: $\widehat{f}_{kNN}(x) = \arg\max_y \ \widehat{p}_{kNN}(x|y)\widehat{P}(y)$

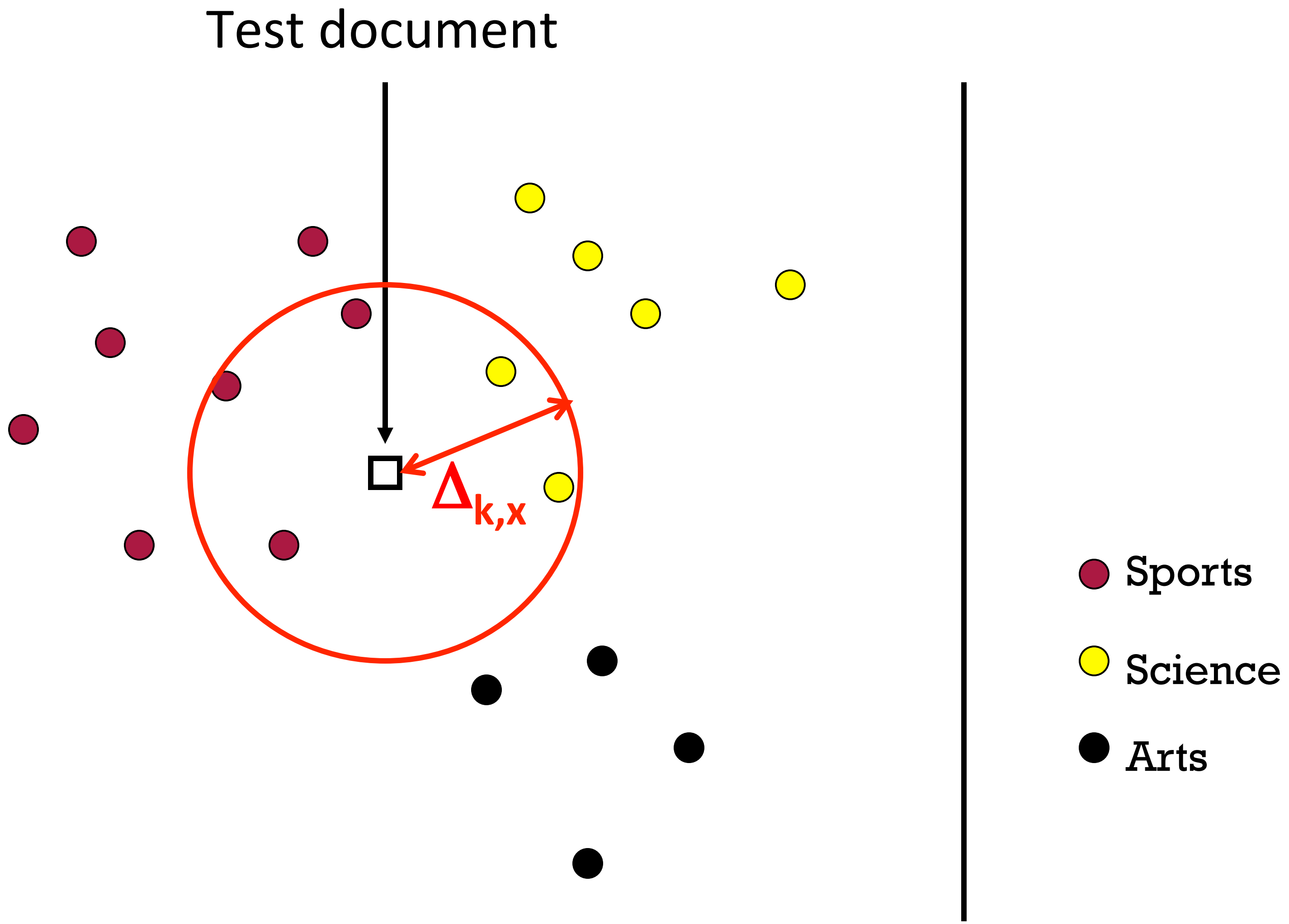
k-NN Density Estimate:

$$\widehat{p}(x) = \frac{k}{n\Delta_{k,x}}$$



k/n is estimated probability in ball of volume $\Delta_{k,x}$

k-NN classifier (k=5)



k-NN classifier

- Optimal Classifier:
$$\begin{aligned} f^*(x) &= \arg\max_y P(y|x) \\ &= \arg\max_y p(x|y)P(y) \end{aligned}$$

- k-NN Classifier:
$$\begin{aligned} \widehat{f}_{kNN}(x) &= \arg\max_y \; \widehat{p}_{kNN}(x|y)\widehat{P}(y) \\ &= \arg\max_y \; k_y \end{aligned}$$

$$\widehat{p}_{kNN}(x|y) = \frac{k_y}{n_y \Delta_{k,x}}$$

training pts of class y that lie within $\Delta_{k,x}$ ball

total training pts of class y

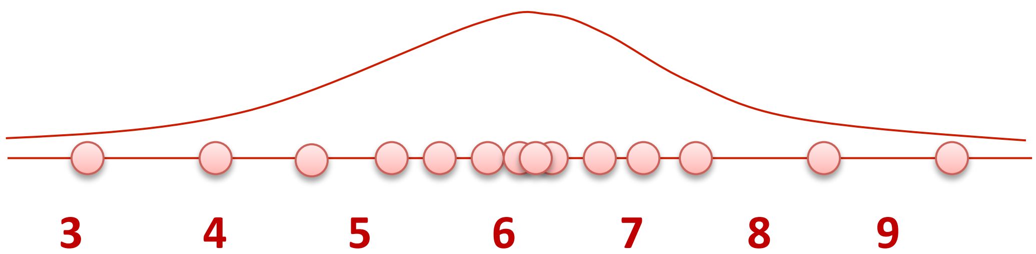
$$\sum_y k_y = k$$

$$\widehat{P}(y) = \frac{n_y}{n}$$

From
Classification
to
Density estimation

Density estimation

Goal: Given $X_1, X_2, \ldots, X_n$, estimate $P(X)$



Distribution of sleep hrs

Distribution of intensities at a pixel

Parametric approaches

Binary X	$P(X) \sim \text{Bernoulli}(\theta)$
Real X	$P(X) \sim \text{Gaussian}(\mu,\sigma)$

Estimate $P(X)$ = estimate parameters θ,μ,σ

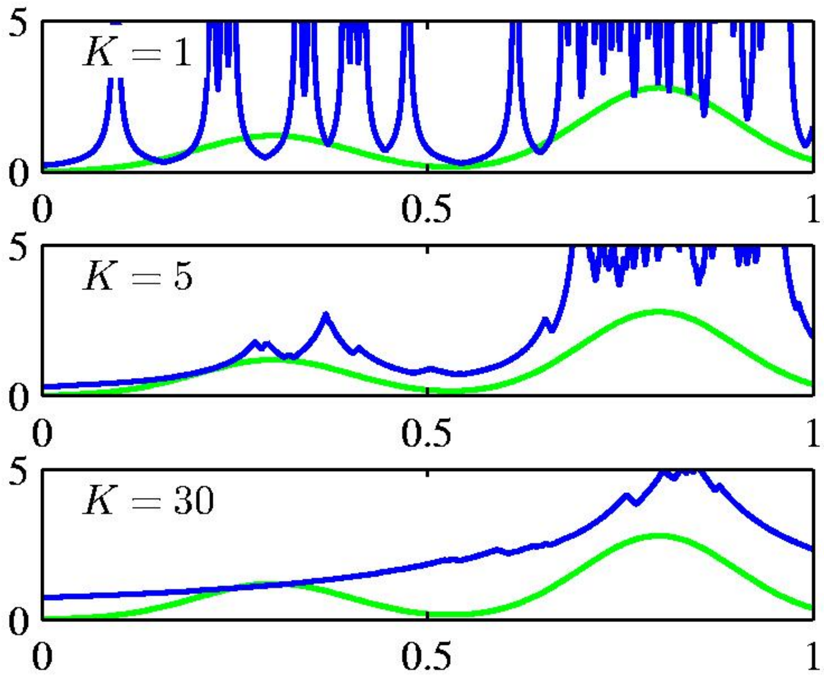
Methods: MLE, MAP

Nonparametric approaches

Methods: k-NN, Histogram and Kernel density estimation

k-NN density estimation

$$\widehat{p}(x) = \frac{k}{n\Delta_{k,x}}$$



k acts as a smoother.

Not very popular for density estimation – spiked estimates

Histogram density estimate

Partition the feature space into distinct bins with widths Δ_i and count the number of observations, n_i, in each bin.

$$\widehat{p}(x) = \frac{n_i}{n\Delta_i}\mathbf{1}_{x\in \mathrm{Bin}_i}$$

"Local relative frequency"

- Often, the same width is used for all bins, $\Delta_i = \Delta$.
- Δ acts as a smoothing parameter.

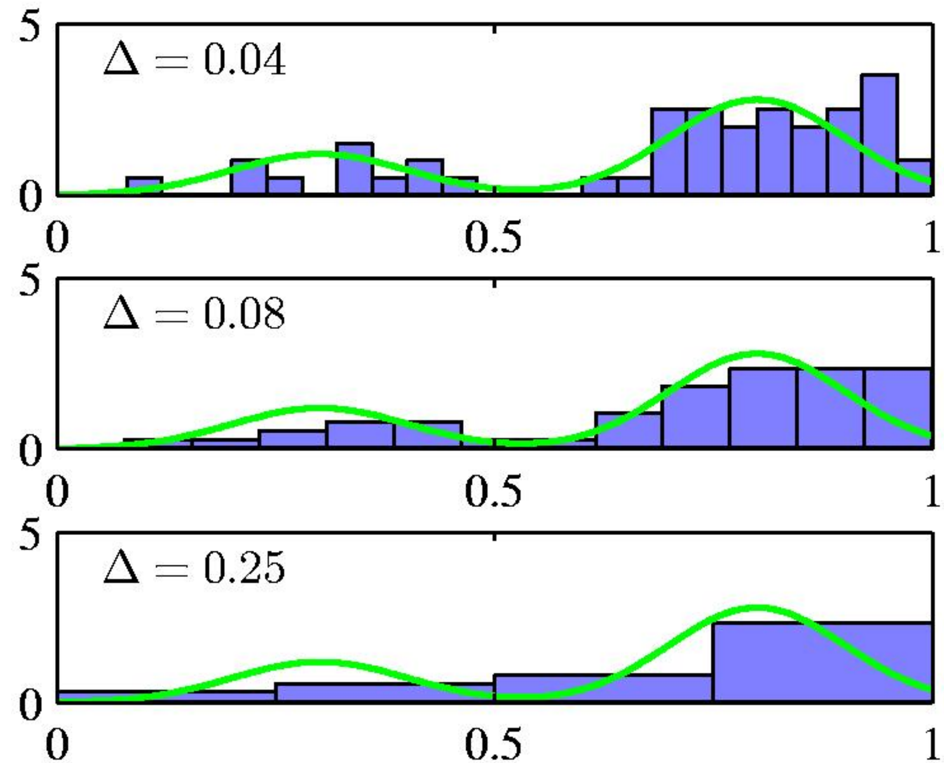


Image src: Bishop book

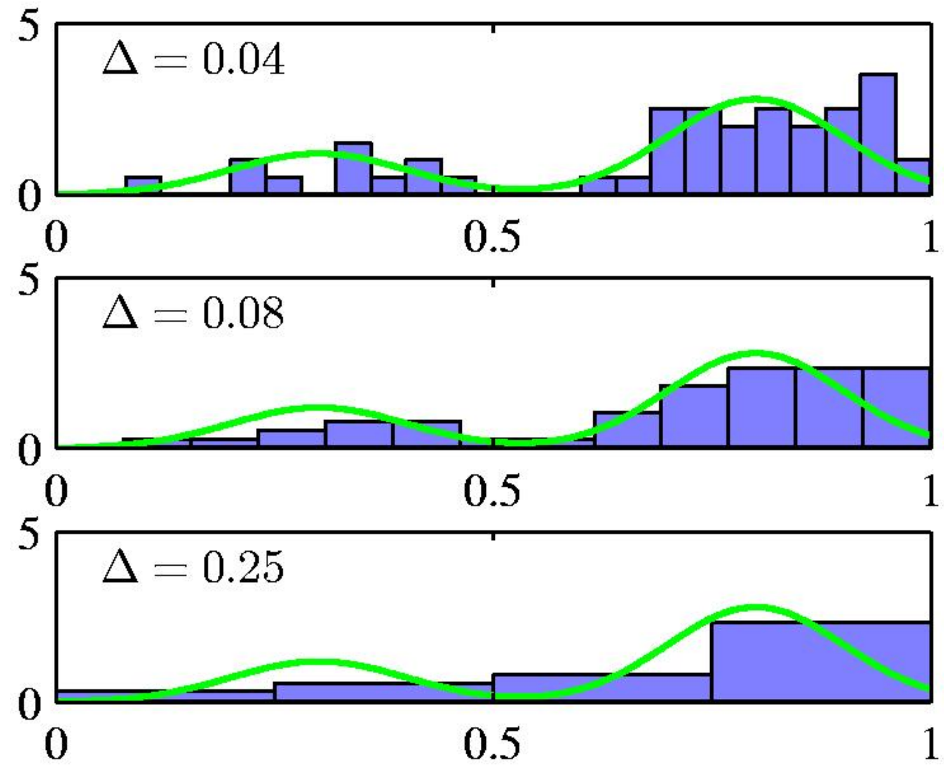
Effect of histogram bin width

$$\widehat{p}(x) = \frac{n_i}{n\Delta}\mathbf{1}_{x\in \mathrm{Bin}_i}$$

bins = 1/Δ

Small Δ, large #bins
Good fit but unstable
(few points per bin)
"**Small bias**, **Large variance**"

Large Δ, small #bins
Poor fit but stable
(many points per bin)
"**Large bias**, **Small variance**"



Histogram as MLE

- Underlying model – density is constant on each bin
  Parameters p_j : density in bin j
  Note $\sum_j p_j = 1/\Delta$ since $\int p(x)dx = 1$

- Maximize likelihood of data under probability model with parameters p_j

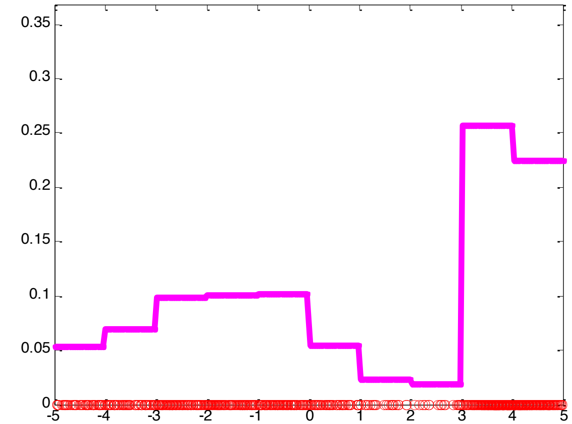
$$\hat{p}(x) = \arg\max_{\{p_j\}} P(X_1, \ldots, X_n; \{p_j\}_{j=1}^{1/\Delta}) \quad \text{s.t.} \quad \sum_j p_j = 1/\Delta$$

- Show that histogram density estimate is MLE under this model

Kernel density estimate

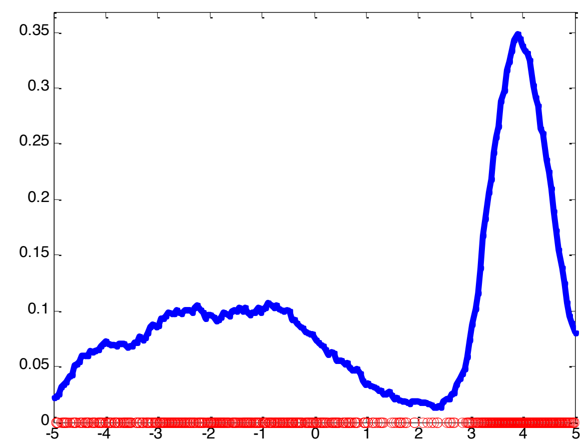
- Histogram – blocky estimate

$$\widehat{p}(x) = \frac{1}{\Delta} \frac{\sum_{j=1}^{n} \mathbf{1}_{X_j \in \text{Bin}_x}}{n}$$



- Kernel density estimate aka "Parzen/moving window method"

$$\widehat{p}(x) = \frac{1}{\Delta} \frac{\sum_{j=1}^{n} \mathbf{1}_{||X_j - x|| \leq \Delta}}{n}$$



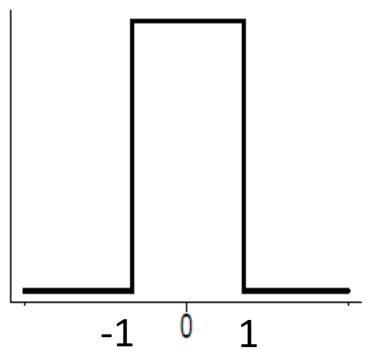
Kernel density estimate

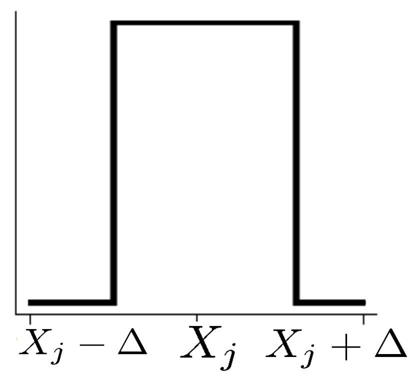
- $\widehat{p}(x) = \frac{1}{\Delta} \frac{\sum_{j=1}^{n} K\left(\frac{X_j - x}{\Delta}\right)}{n}$ more generally

$$K\left(\frac{X_j - x}{\Delta}\right)$$

boxcar kernel :

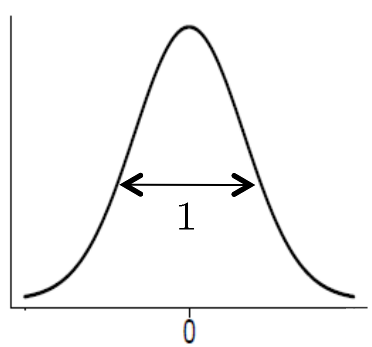
$$K(x) = \frac{1}{2} I(x),$$

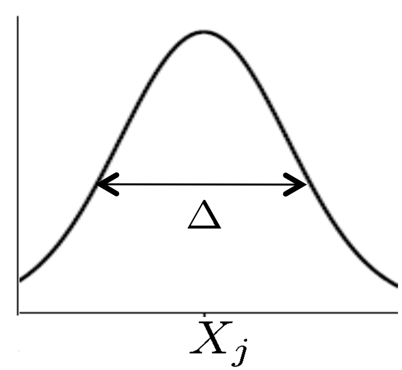




Gaussian kernel :

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

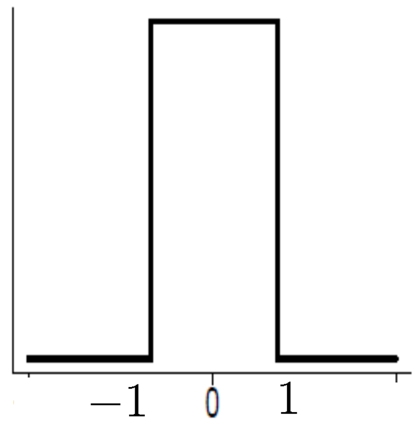




Kernels

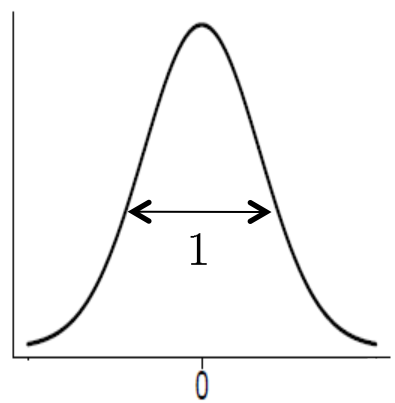
boxcar kernel :

$$K(x) = \frac{1}{2}I(x),$$



Gaussian kernel :

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



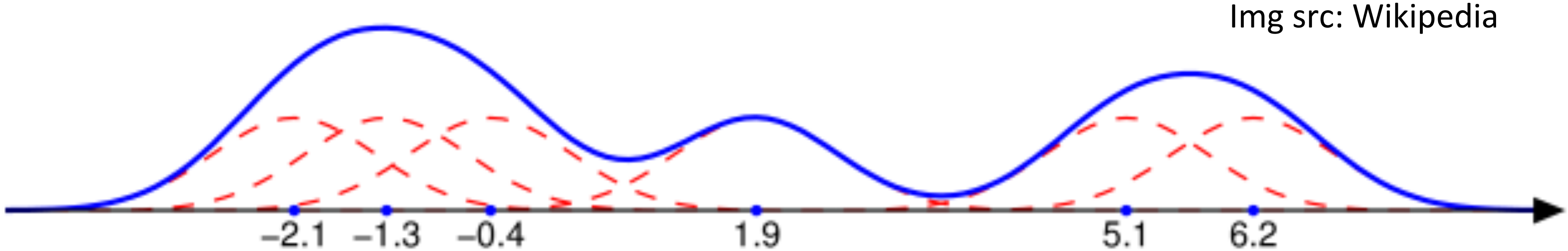
Any kernel function that satisfies

$$\begin{aligned} K(x) &\geq 0, \\ \int K(x)dx &= 1 \end{aligned}$$

Kernel density estimation

- Place small "bumps" at each data point, determined by the kernel function.
- The estimator consists of a (normalized) "sum of bumps”.

Img src: Wikipedia



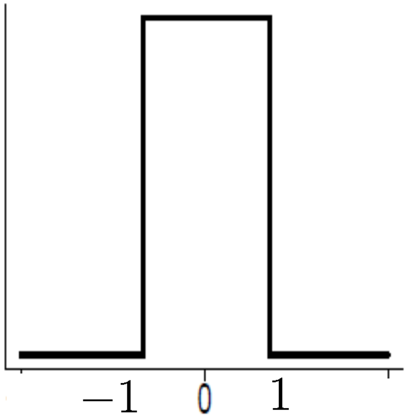
Gaussian bumps (red) around six data points and their sum (blue)

- Note that where the points are denser the density estimate will have higher values.

Choice of Kernels

boxcar kernel :

$$K(x) = \frac{1}{2} I(x),$$

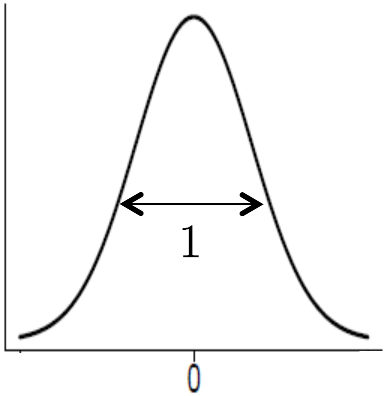


Finite support

– only need local points to compute estimate

Gaussian kernel :

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



Infinite support

- need all points to compute estimate
-But quite popular since smoother

Choice of kernel bandwidth

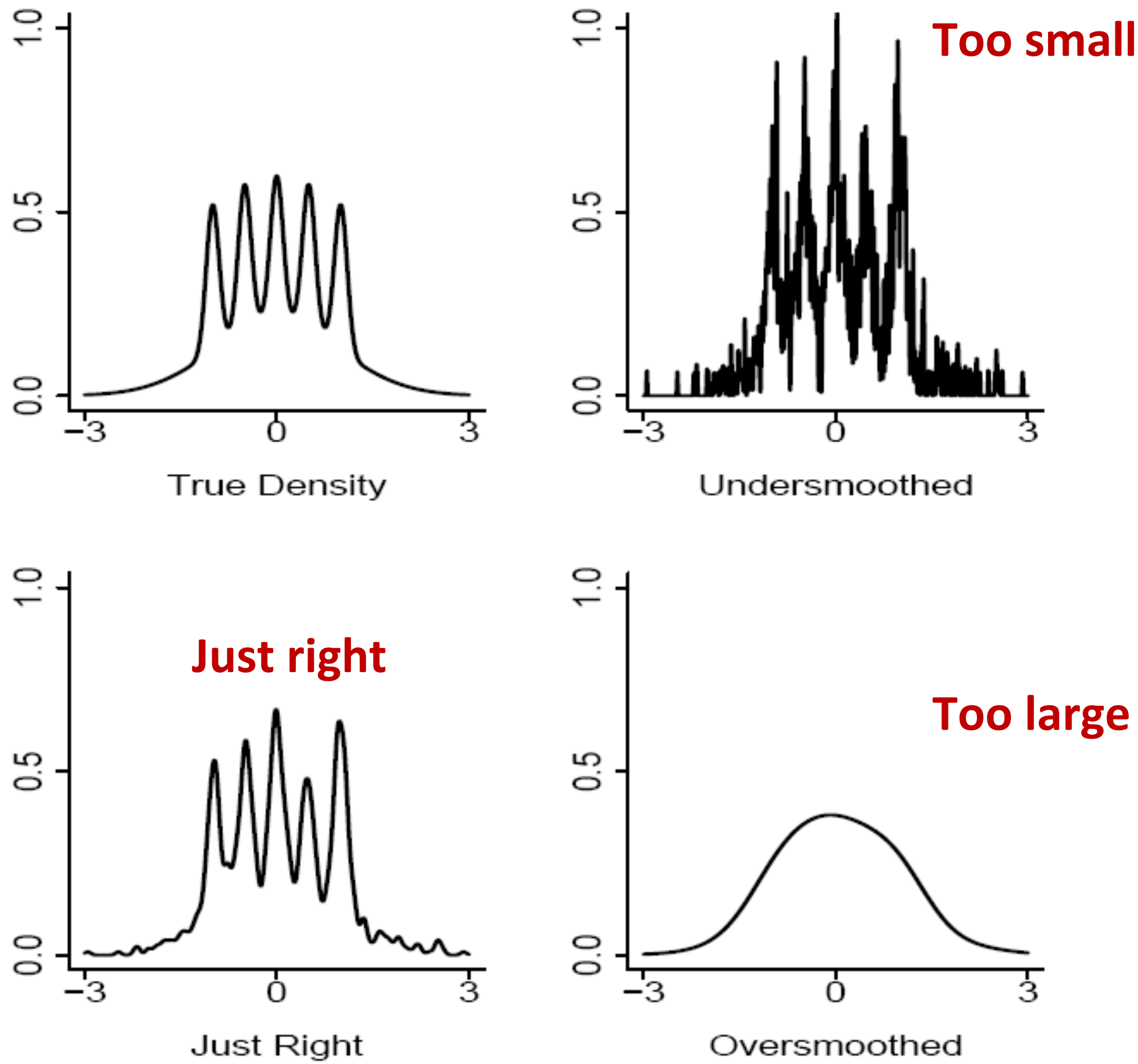
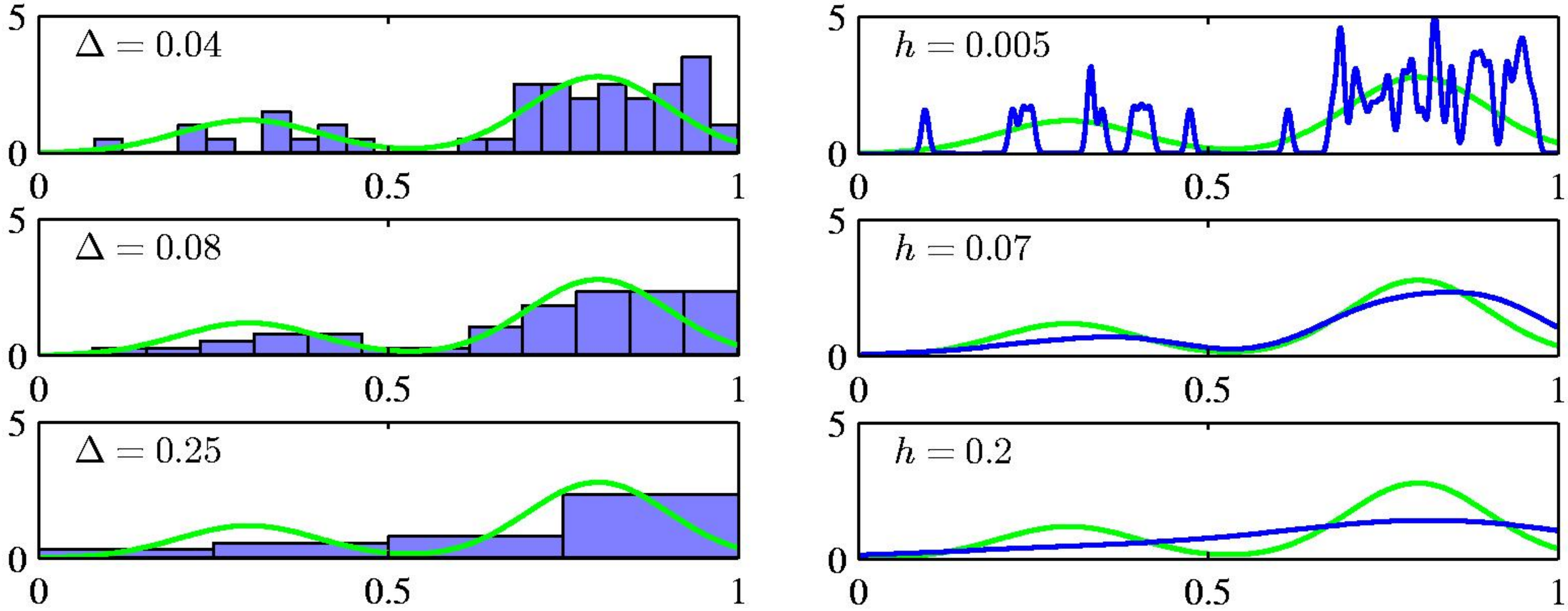


Image Source: Larry's book – All of Nonparametric Statistics

Bart-Simpson Density

Histograms vs. Kernel density estimation



Δ = h acts as a smoother.

Nonparametric density estimation

- Histogram $$\widehat{p}(x) = \frac{n_i}{n\Delta}\mathbf{1}_{x\in \mathrm{Bin}_i}$$
- Kernel density est $$\widehat{p}(x) = \frac{n_x}{n\Delta}$$

Fix Δ, estimate number of points within Δ of x (n_i or n_x) from data

Fix $n_x = k$, estimate Δ from data (volume of ball around x that contains k training pts)

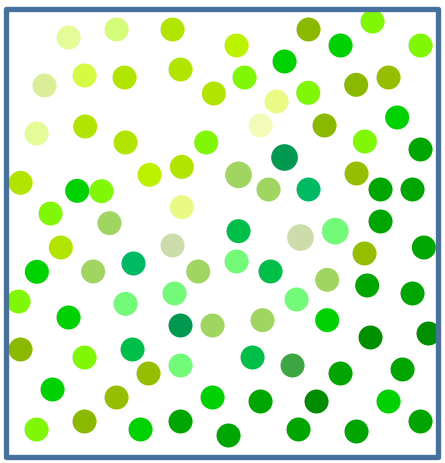
- k-NN density est $$\widehat{p}(x) = \frac{k}{n\Delta_{k,x}}$$

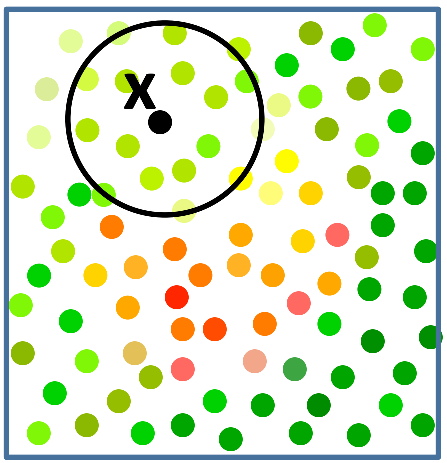
From Classification and Density Estimation to Regression

Temperature sensing

- What is the temperature in the room?

at location x?





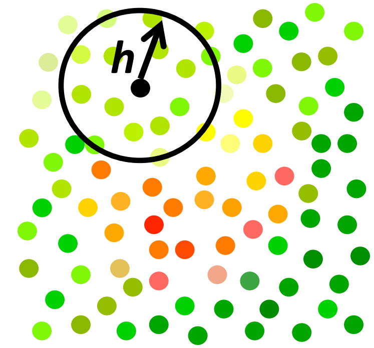
$$\widehat{T} = \frac{1}{n} \sum_{i=1}^{n} Y_i$$

Average

$$\widehat{T}(x) = \frac{\sum_{i=1}^{n} Y_i \mathbf{1}_{||X_i - x|| \leq h}}{\sum_{i=1}^{n} \mathbf{1}_{||X_i - x|| \leq h}}$$

"Local" Average

Kernel Regression



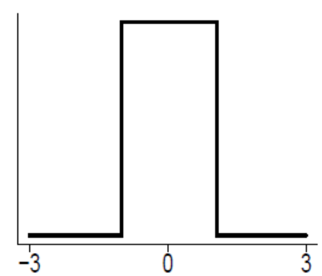
- Aka Local Regression
- Nadaraya-Watson Kernel Estimator

$$\widehat{f}_n(X) = \sum_{i=1}^{n} w_i Y_i \quad \text{Where} \quad w_i(X) = \frac{K\left(\frac{X-X_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{X-X_i}{h}\right)}$$

- Weight each training point based on distance to test point
- Boxcar kernel yields local average

boxcar kernel :

$$K(x) = \frac{1}{2} I(x),$$

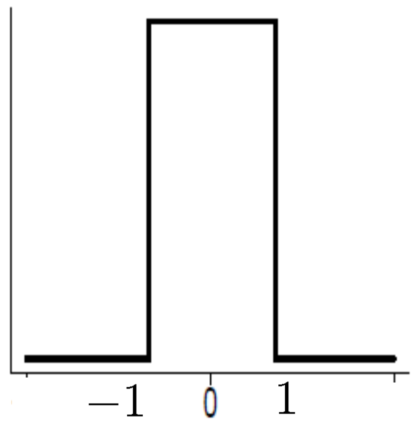


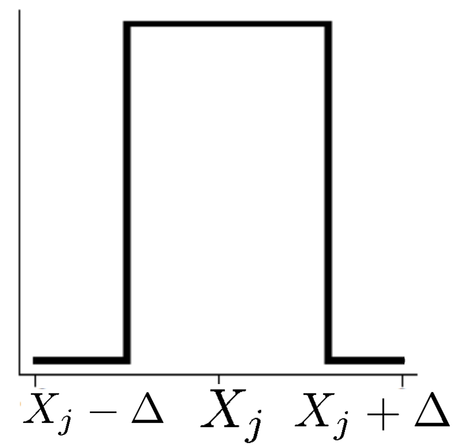
Kernels

$$K\left(\frac{X_j - x}{\Delta}\right)$$

boxcar kernel :

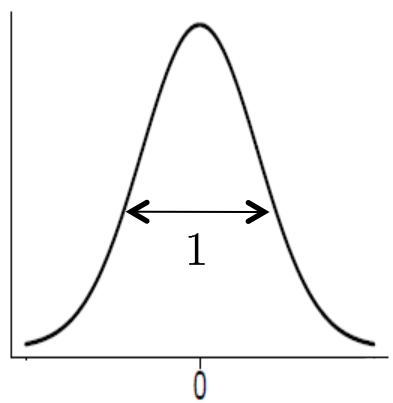
$$K(x) = \frac{1}{2}I(x),$$

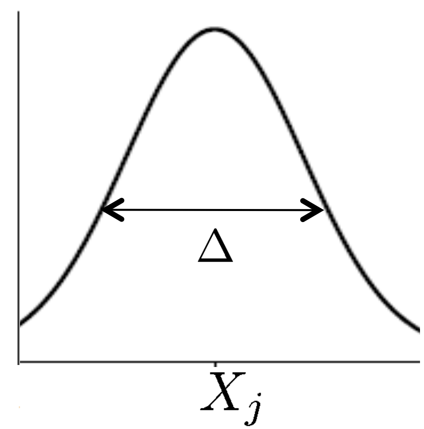




Gaussian kernel :

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





Choice of kernel bandwidth h

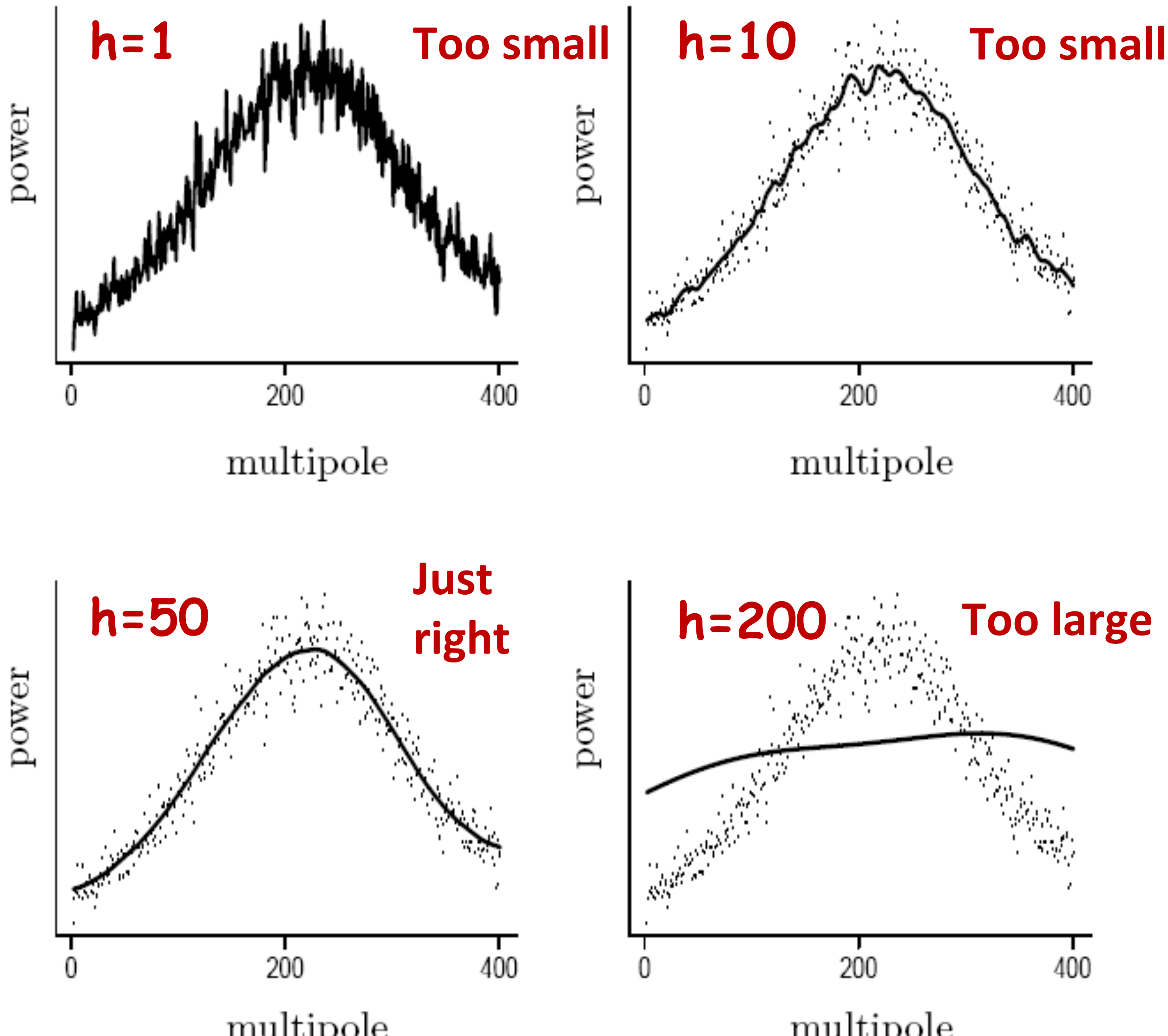


Image Source: Larry's book – All of Nonparametric Statistics

Choice of kernel is not that important

Kernel Regression as Weighted Least Squares

$$\min_f \sum_{i=1}^{n} w_i (f(X_i) - Y_i)^2 \qquad\qquad w_i(X) = \frac{K\left(\frac{X-X_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{X-X_i}{h}\right)}$$

Weighted Least Squares

Kernel regression corresponds to locally constant estimator obtained from (locally) weighted least squares

i.e. set $f(X_i) = \beta$ (a constant)

Kernel Regression as Weighted Least Squares

set $f(X_i) = \beta$ (a constant)

$$\min_{\beta} \sum_{i=1}^{n} w_i(\underset{\text{constant}}{\underset{\downarrow}{\beta}} - Y_i)^2 \qquad\qquad w_i(X) = \frac{K\left(\frac{X-X_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{X-X_i}{h}\right)}$$

$$\frac{\partial J(\beta)}{\partial \beta} = 2\sum_{i=1}^{n} w_i(\beta - Y_i) = 0$$

Notice that $\sum_{i=1}^{n} w_i = 1$

$$\Rightarrow \widehat{f}_n(X) = \widehat{\beta} = \sum_{i=1}^{n} w_i Y_i$$

Local Linear/Polynomial Regression

$$\min_f \sum_{i=1}^{n} w_i(f(X_i) - Y_i)^2 \qquad\qquad w_i(X) = \frac{K\left(\frac{X-X_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{X-X_i}{h}\right)}$$

Weighted Least Squares

Local Polynomial regression corresponds to locally polynomial estimator obtained from (locally) weighted least squares

i.e. set

$$f(X_i) = \beta_0 + \beta_1(X_i - X) + \frac{\beta_2}{2!}(X_i - X)^2 + \cdots + \frac{\beta_p}{p!}(X_i - X)^p$$

(local polynomial of degree p around X)

Summary

- Non-parametric approaches

Four things make a nonparametric/memory/instance based/lazy learner:

1. *A distance metric, dist(x,X_i)*
   Euclidean (and many more)
2. *How many nearby neighbors/radius to look at?*
   k, Δ/h
3. *A weighting function (optional)*
   W based on kernel K
4. *How to fit with the local points?*
   Average, Majority vote, Weighted average, Poly fit

Summary

- Parametric vs Nonparametric approaches
  - ➢ Nonparametric models place very mild assumptions on the data distribution and provide good models for complex data

    Parametric models rely on very strong (simplistic) distributional assumptions
  - ➢ Nonparametric models (not histograms) requires storing and computing with the entire data set.

    Parametric models, once fitted, are much more efficient in terms of storage and computation.