

CSCI 4360/6360 Data Science II

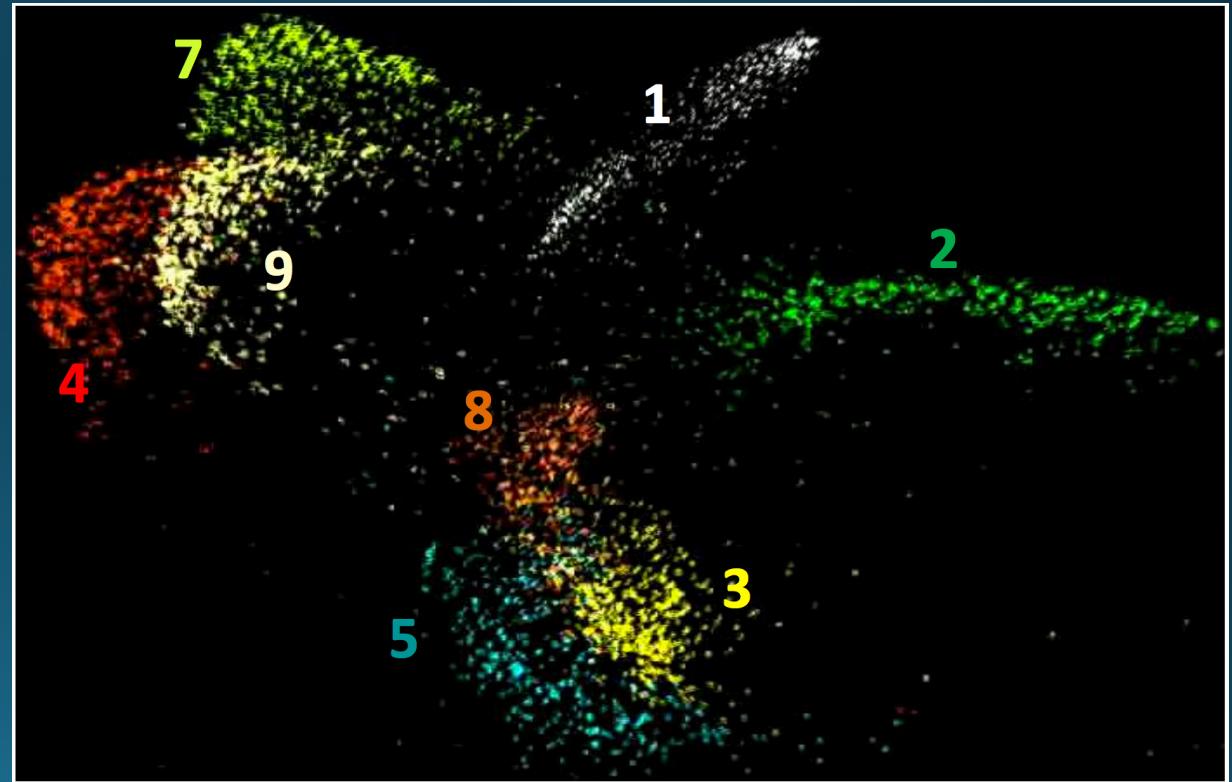
# Kernel Methods

# Parametric Statistics

- Assume some functional form (Gaussian, Bernoulli, Multinomial, logistic, linear) for
  - $P(X_i|Y)$  and  $P(Y)$  as in Naïve Bayes
  - $P(Y|X)$  as in Logistic Regression
- Estimate parameters ( $\mu, \sigma^2, \theta, w, \beta$ ) using MLE/MAP
  - Plug-n-chug
- **Advantages:** need relatively few data points to learn parameters
- **Drawbacks:** Strong assumptions rarely satisfied in practice

# Embeddings

- Again!
- MNIST, projected into 2D embedding space
- What distribution do these follow?
- **Highly nonlinear**

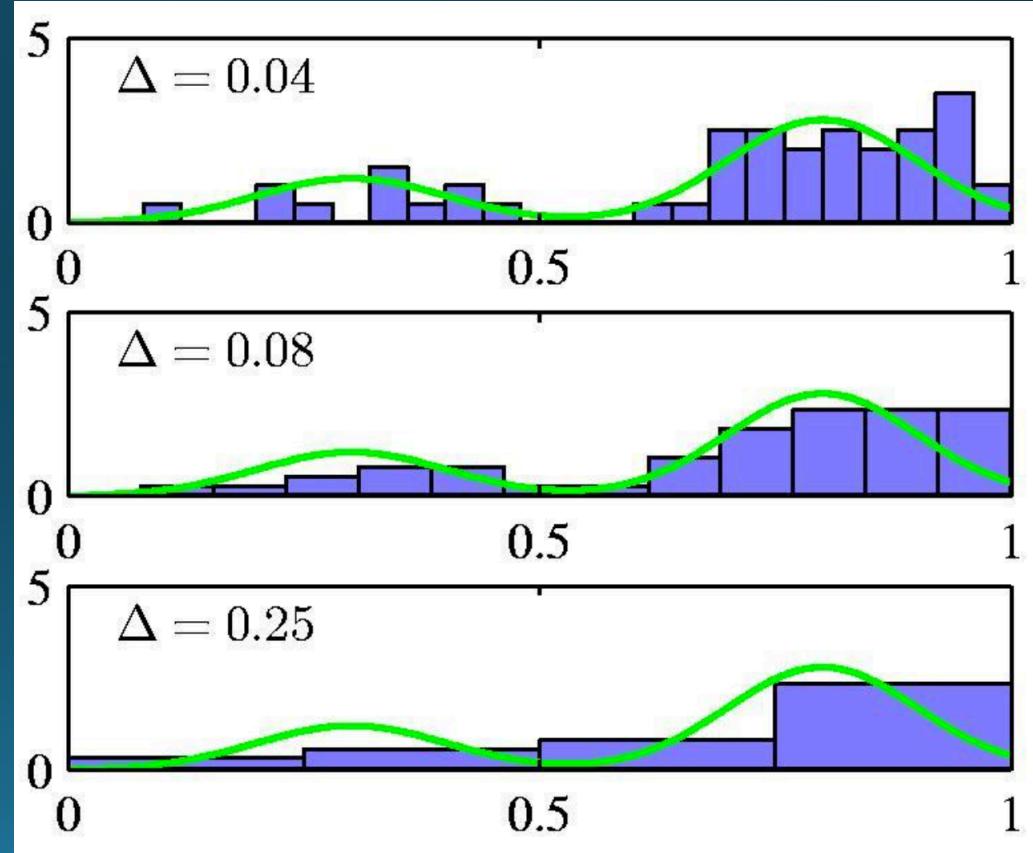


# Nonparametric Statistics

- Typically very few, if any, distributional assumptions
- Usually requires more data
- Let number of parameters scale with the data
- **Today**
  - Kernel density estimation
  - K-nearest neighbors classification
  - Kernel regression

# Density Estimation

- You've done this before—histograms!
- Partition feature space into distinct bins with specified widths and count number of observations  $n_i$  in each bin
$$\hat{p}(x) = \frac{n_i}{n\Delta_i} \mathbf{1}_{x \in \text{Bin}_i}$$
- Same width is often used for all bins
- Bin width acts as **smoothing parameter**



# Effect of $\Delta$

- # of bins =  $1/\Delta$

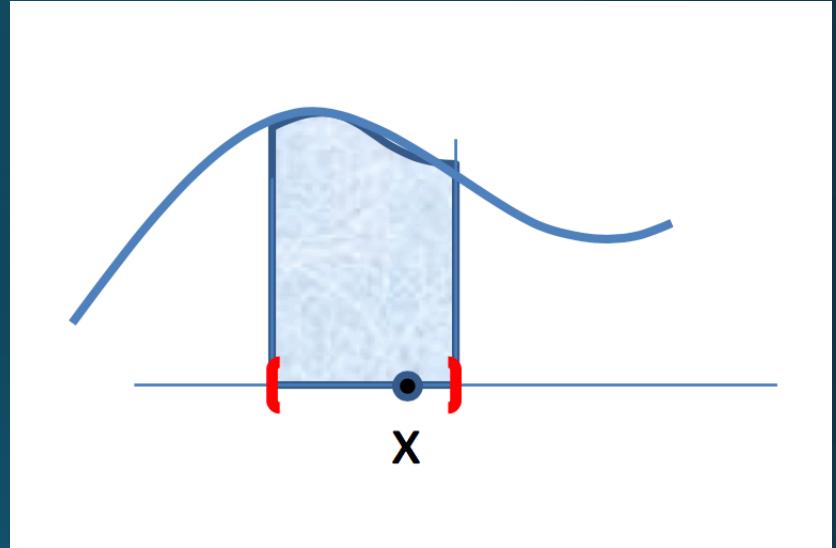
$$\hat{p}(x) = \frac{n_i}{n\Delta} \mathbf{1}_{x \in \text{Bin}_i}$$

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

- Bias of histogram density estimate

$$\begin{aligned} \mathbb{E}[\hat{p}(x)] &= \frac{1}{\Delta} P(X \in \text{Bin}_x) = \frac{1}{\Delta} \int_{z \in \text{Bin}_x} p(z) dz \approx \frac{p(x)\Delta}{\Delta} \\ &= 1 \end{aligned}$$

Assuming density is roughly constant in each bin  
(roughly true, if  $\Delta$  is small)



# Bias-Variance Trade-off

- Choice of # of bins
  - if  $\Delta$  is small
  - if  $\Delta$  is large

$$\mathbb{E} [\hat{p}(x)] \approx p(x)$$
$$\mathbb{E} [\hat{p}(x)] \approx \hat{p}(x)$$

$p(x)$  approximately constant per bin

More data per bin stabilizes estimate

- **Bias:** how close is mean of estimate to the truth
- **Variance:** how much does estimate vary around the mean

Small  $\Delta$ , large #bins



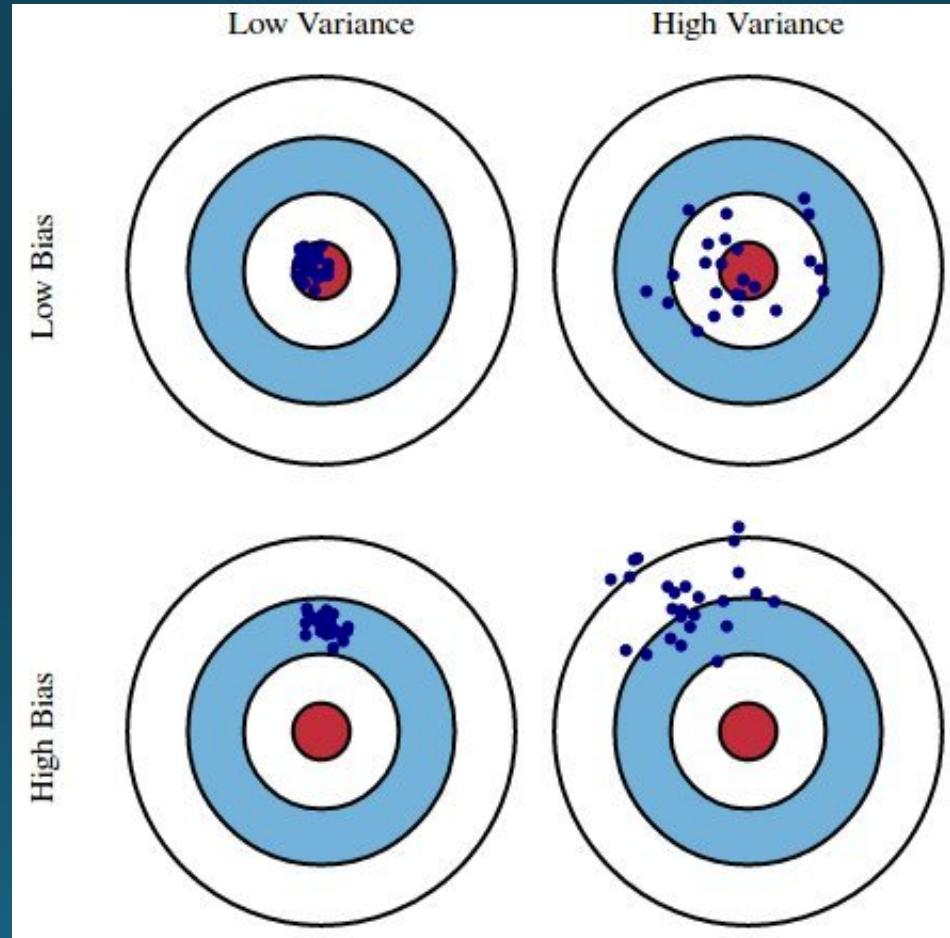
**“Small bias, Large variance”**

Large  $\Delta$ , small #bins



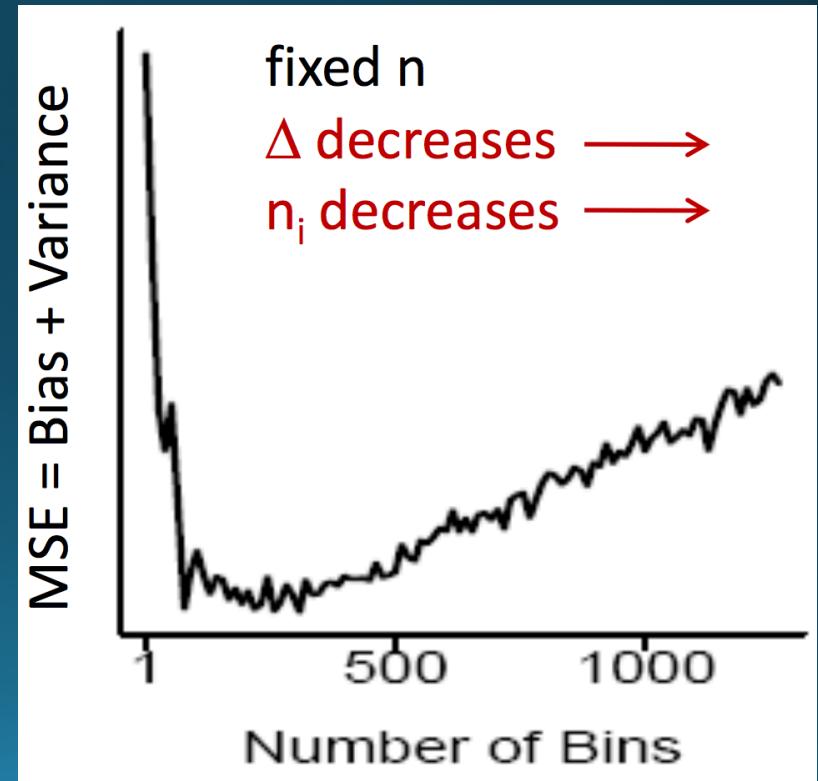
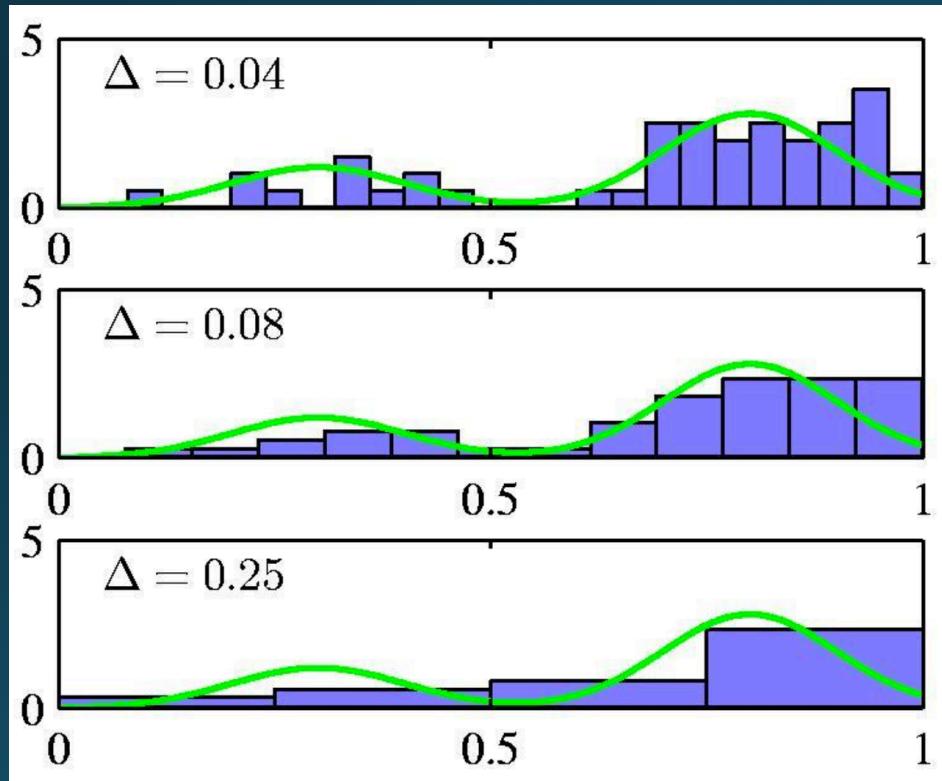
**“Large bias, Small variance”**

# Bias-Variance Trade-off



# Choice of number of bins

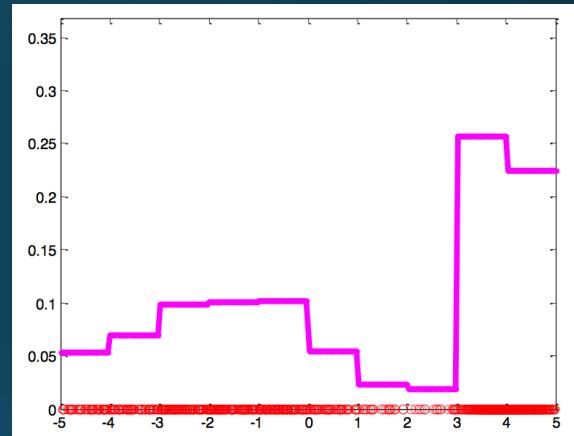
$$\hat{p}(x) = \frac{n_i}{n\Delta} \mathbf{1}_{x \in \text{Bin}_i}$$



# Kernel Density Estimation

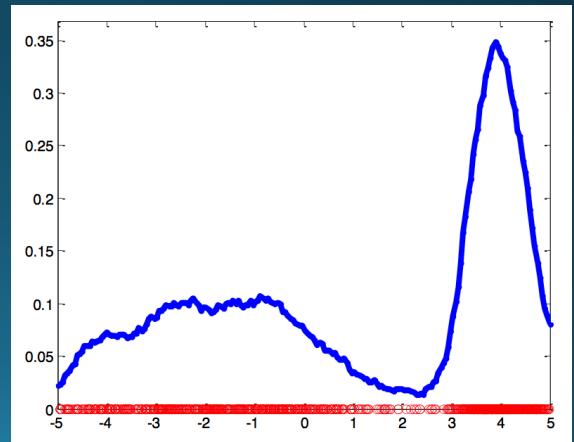
- Histograms are “blocky” estimates

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



- Kernel density estimate, aka “Parzen / moving window” method

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



# Kernel Density Estimation

- More generally:

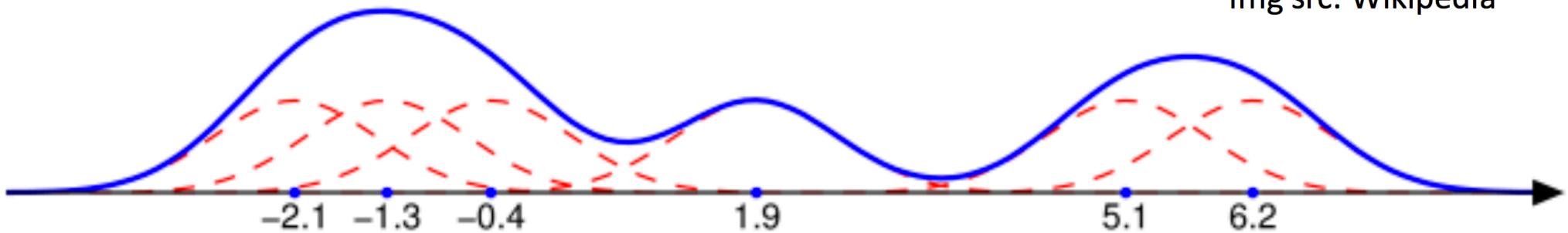
$$\hat{p}(x) = \frac{1}{\Delta} \frac{\sum_{j=1}^n K\left(\frac{X_j - x}{\Delta}\right)}{n}$$

- $K$  is the kernel function (not to be confused with  $\varphi$  from Kernel PCA)
- Embodies any number of possible kernel functions

# Kernel Density Estimation

- Places small “bumps” at each data point, determined by  $K$
- Estimator itself consists of a [normalized] “sum of bumps”

Img src: Wikipedia



- Where points are denser, density estimate will be higher

# Kernels

- Any function that satisfies

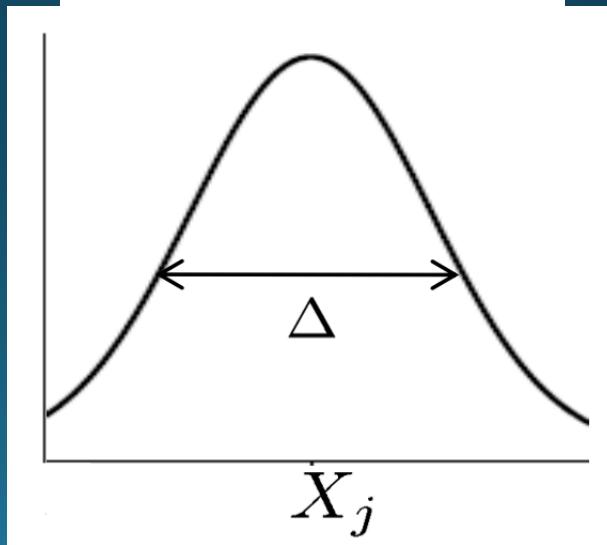
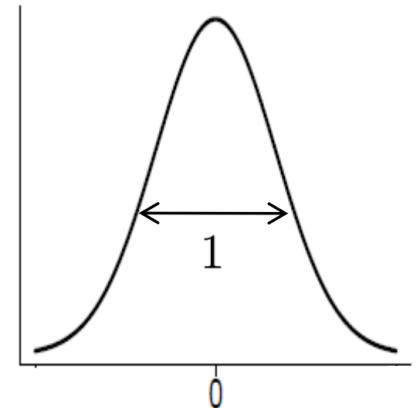
$$K(x) \geq 0$$

$$\int K(x)dx = 1$$

- SciPy has a **ton**
  - See “signal.get\_window”

Gaussian kernel :

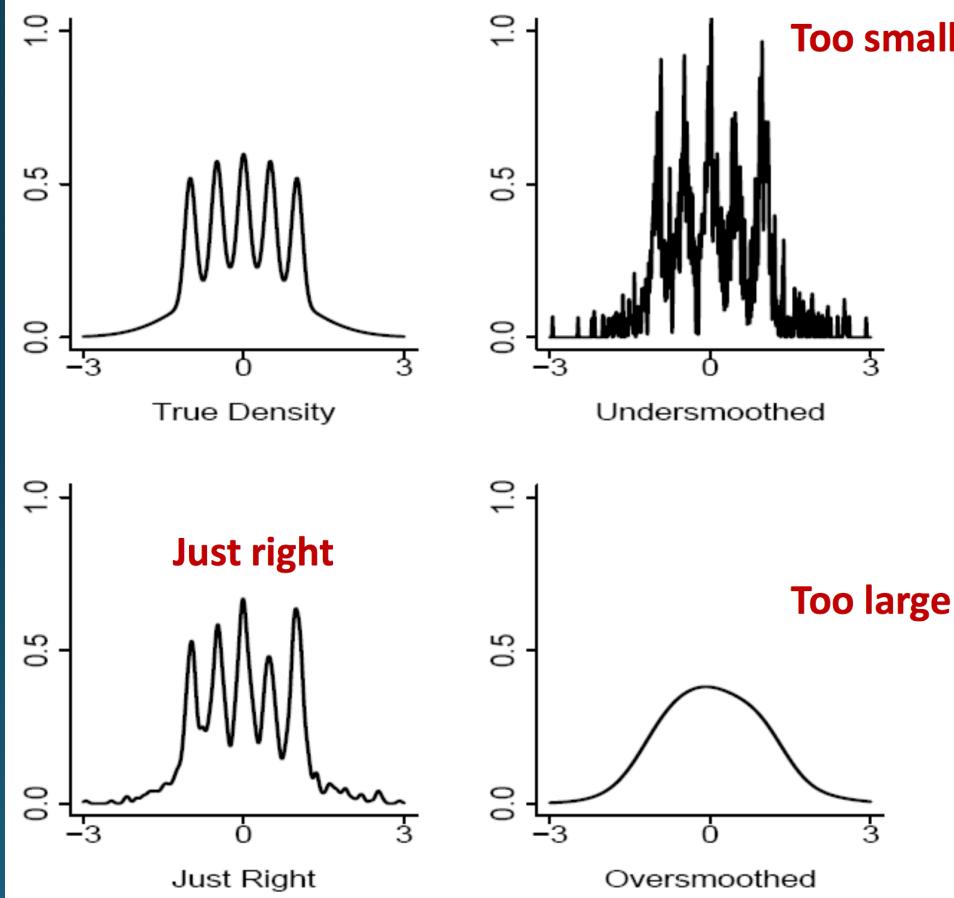
$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$



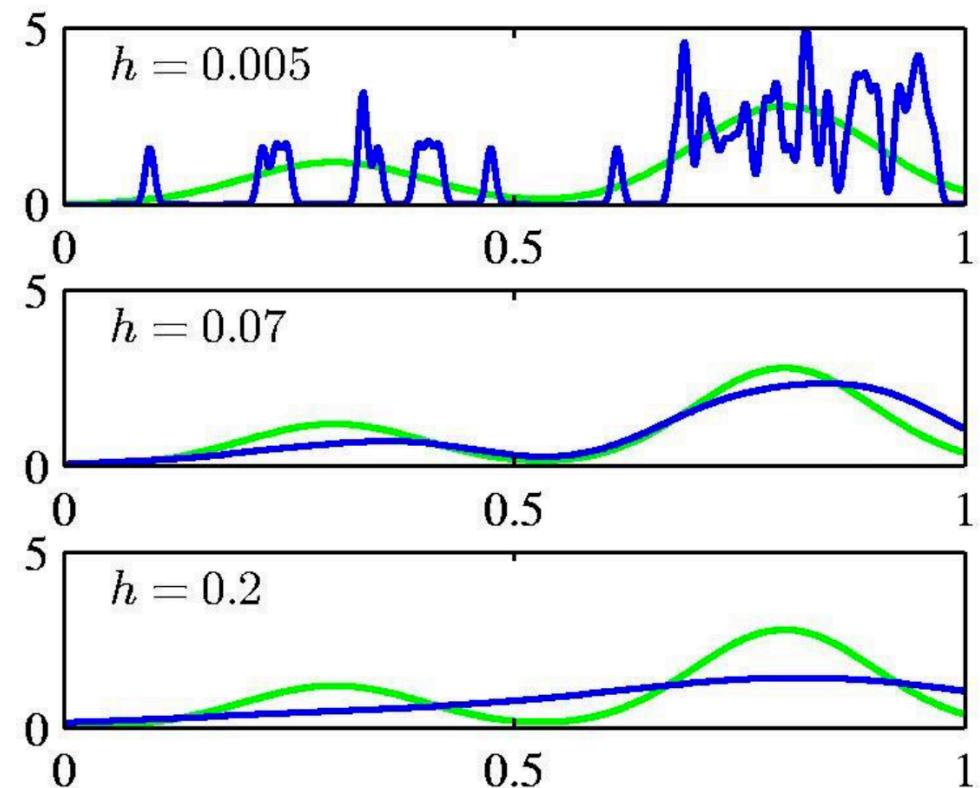
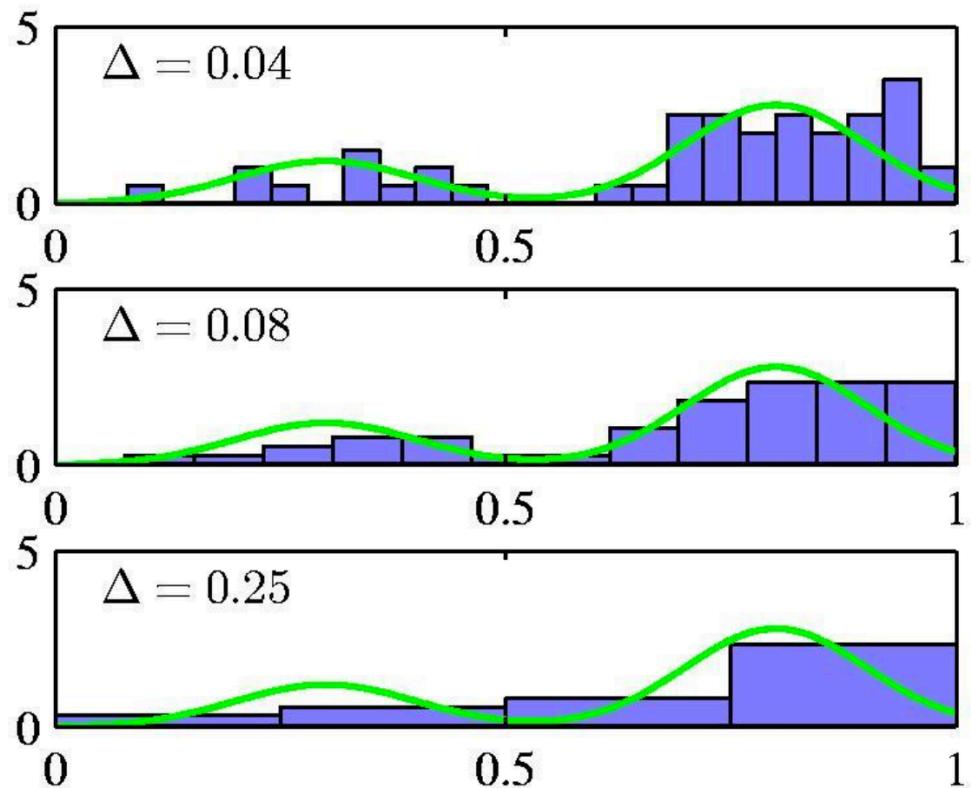
Infinite support: need all points to compute estimate. **But quite popular.**

# Choice of kernel bandwidth

The Bart-Simpson Density



# Histograms versus KDE



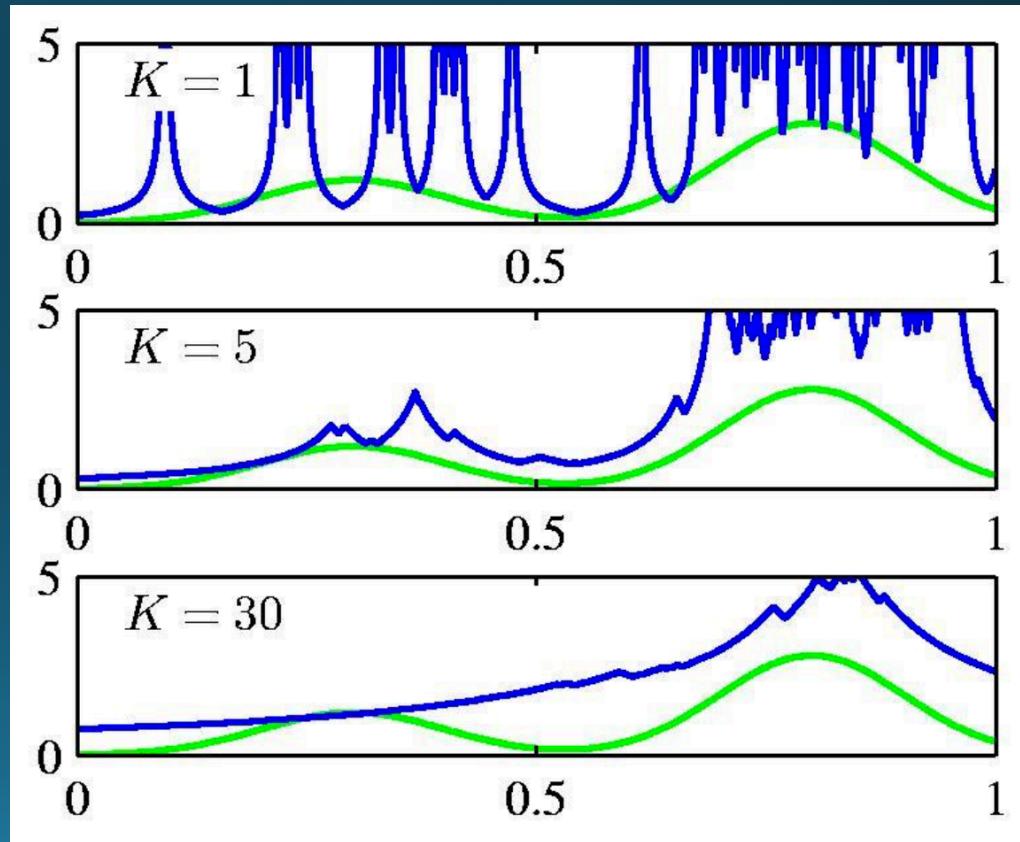
# KNN Density Estimation

- Recall
  - Histograms
  - KDE
- Fix  $\Delta$ , estimate number of points within  $\Delta$  of  $x$  ( $n_i$  or  $n_x$ ) from the data
- Fix  $n_x = k$ , estimate  $\Delta$  from data (volume of ball around  $x$  with  $k$  data points)

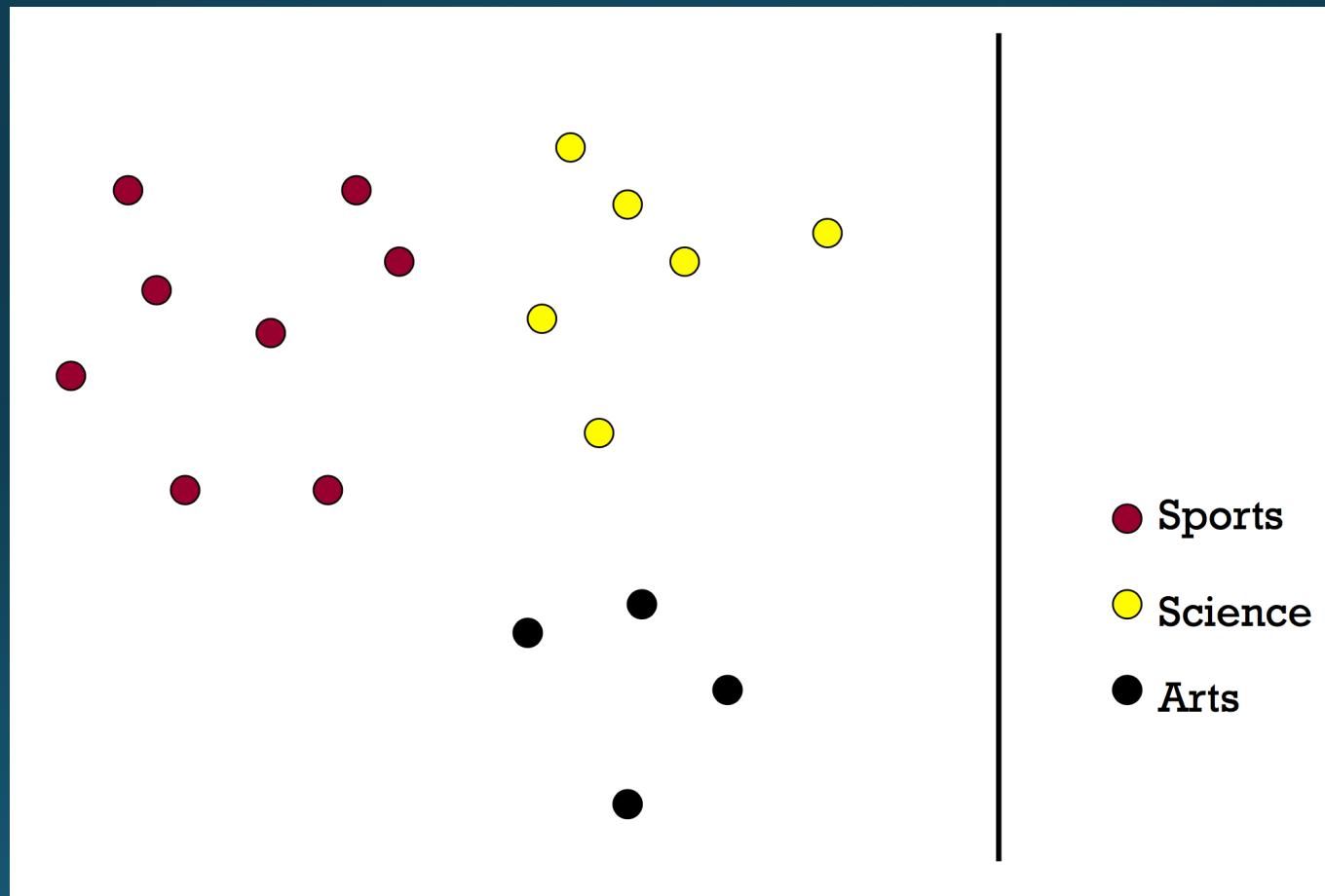
- **KNN Density Estimation** 
$$\hat{p}(x) = \frac{k}{n\Delta_{k,x}}$$

# KNN Density Estimation

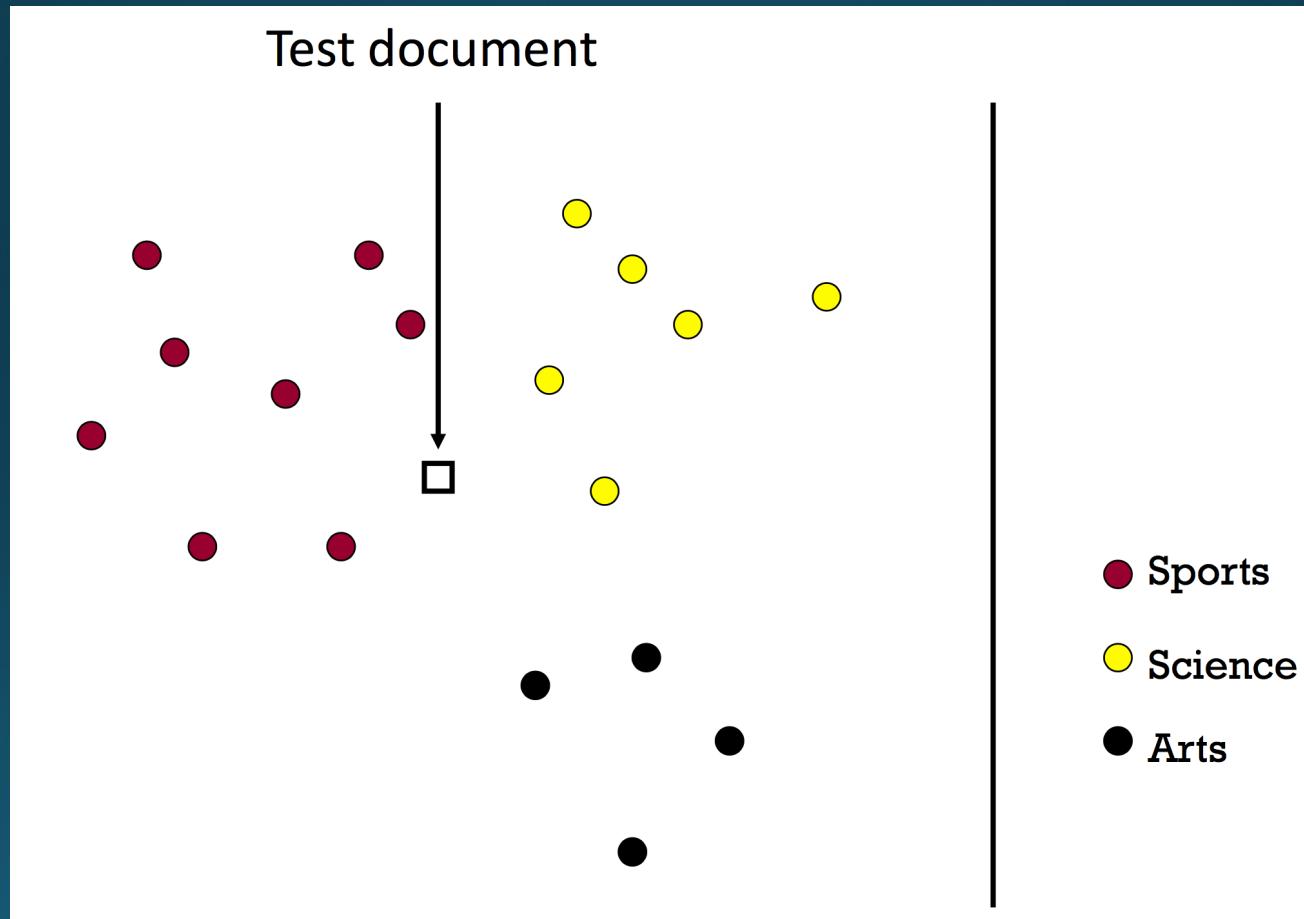
- $k$  acts as a smoother
- Not very popular for density estimation
  - Computationally expensive
  - Estimates are poor
- **But related version for classification is very popular**



# KNN Classification

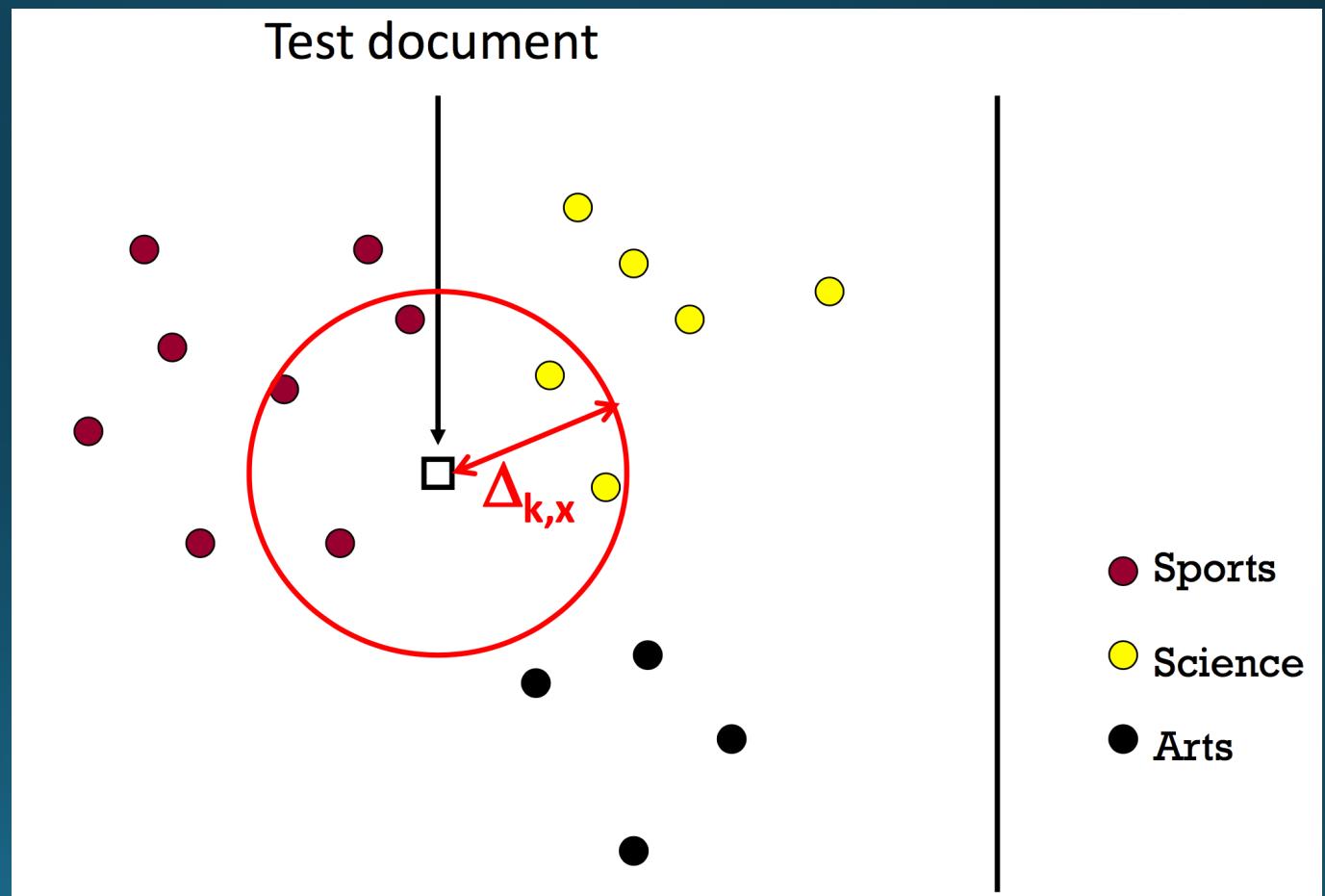


# KNN Classification



# KNN Classification

- $k = 4$
- What should we predict?
- Average?  
Majority? **Why?**



# KNN Classification

- Optimal classifier

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

- KNN classifier

$$\begin{aligned}\hat{f}_{kNN}(x) &= \arg \max_y \hat{p}_{kNN}(x|y)\hat{P}(y) \\&= \arg \max_y k_y\end{aligned}$$

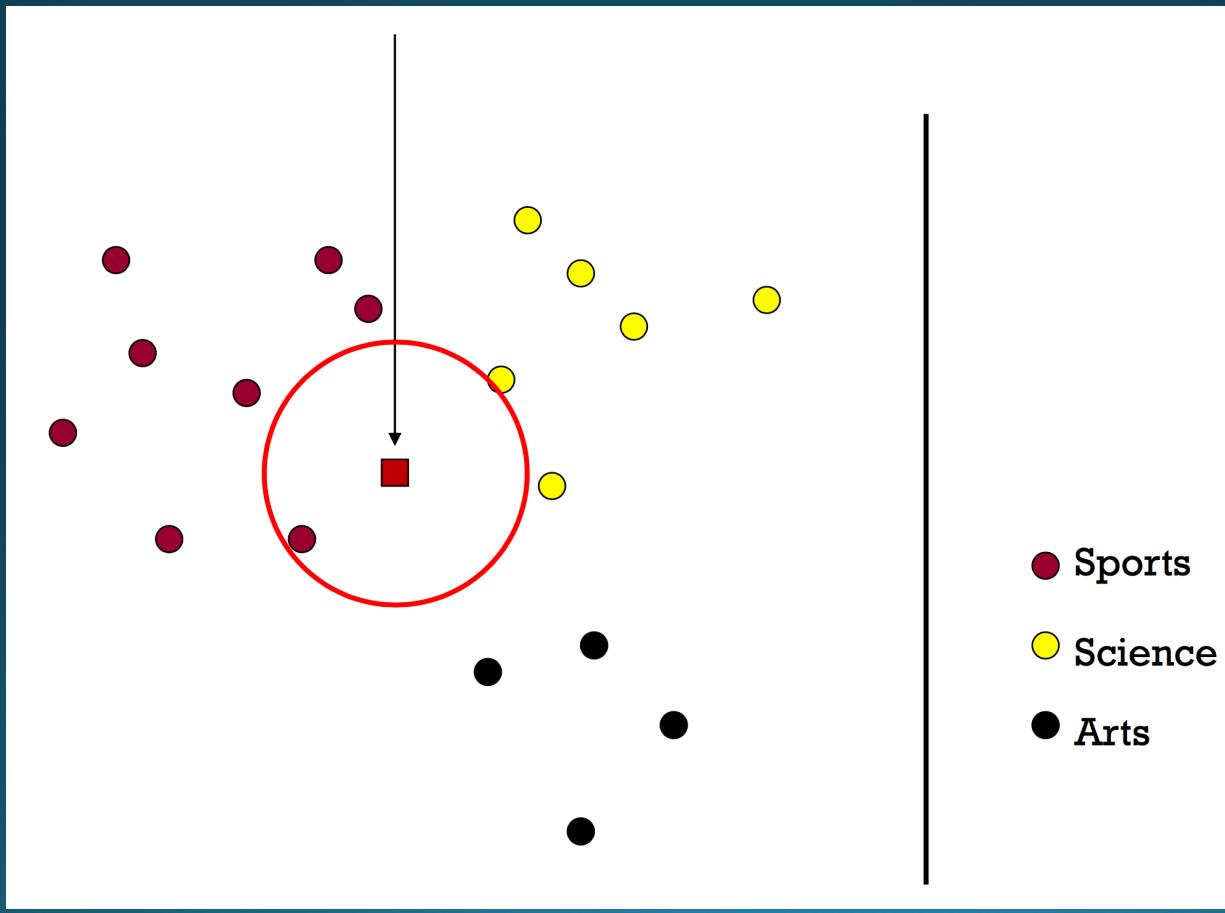
# of training points in class  $y$

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

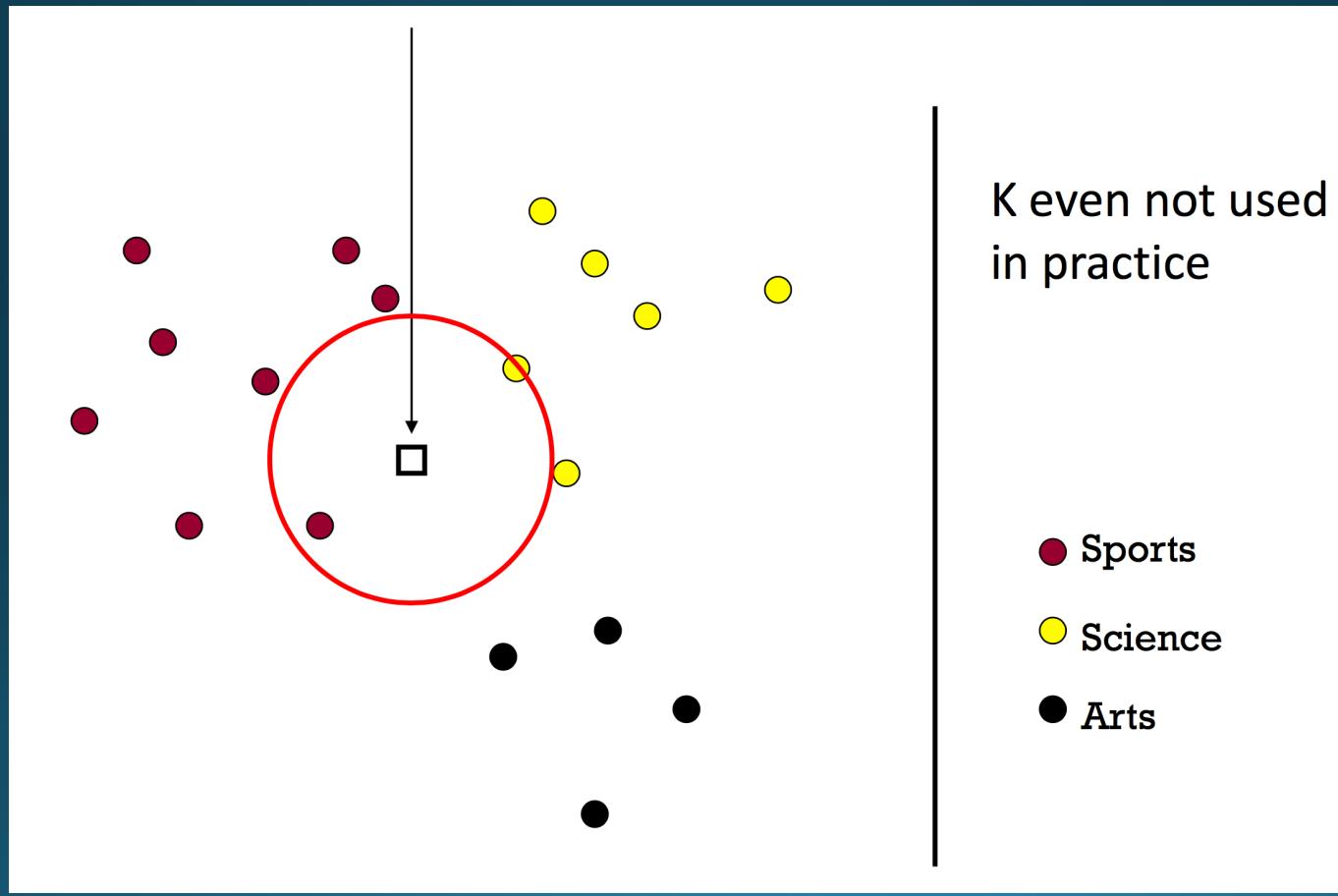
# of training points in class  $y$  that lie within  $\Delta_k$  ball

$$\sum_y k_y = k \quad \hat{P}(y) = \frac{n_y}{n}$$

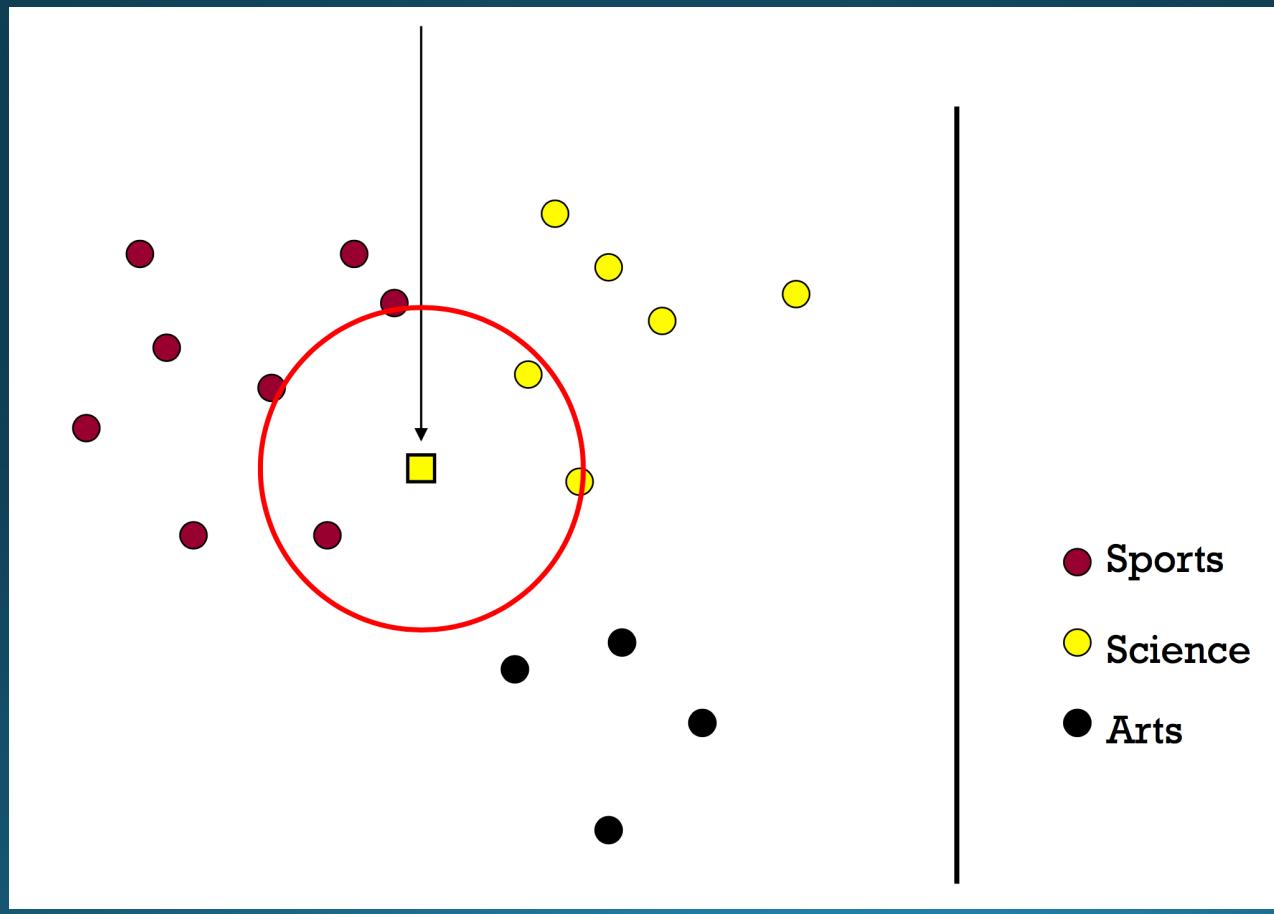
# 1-NN



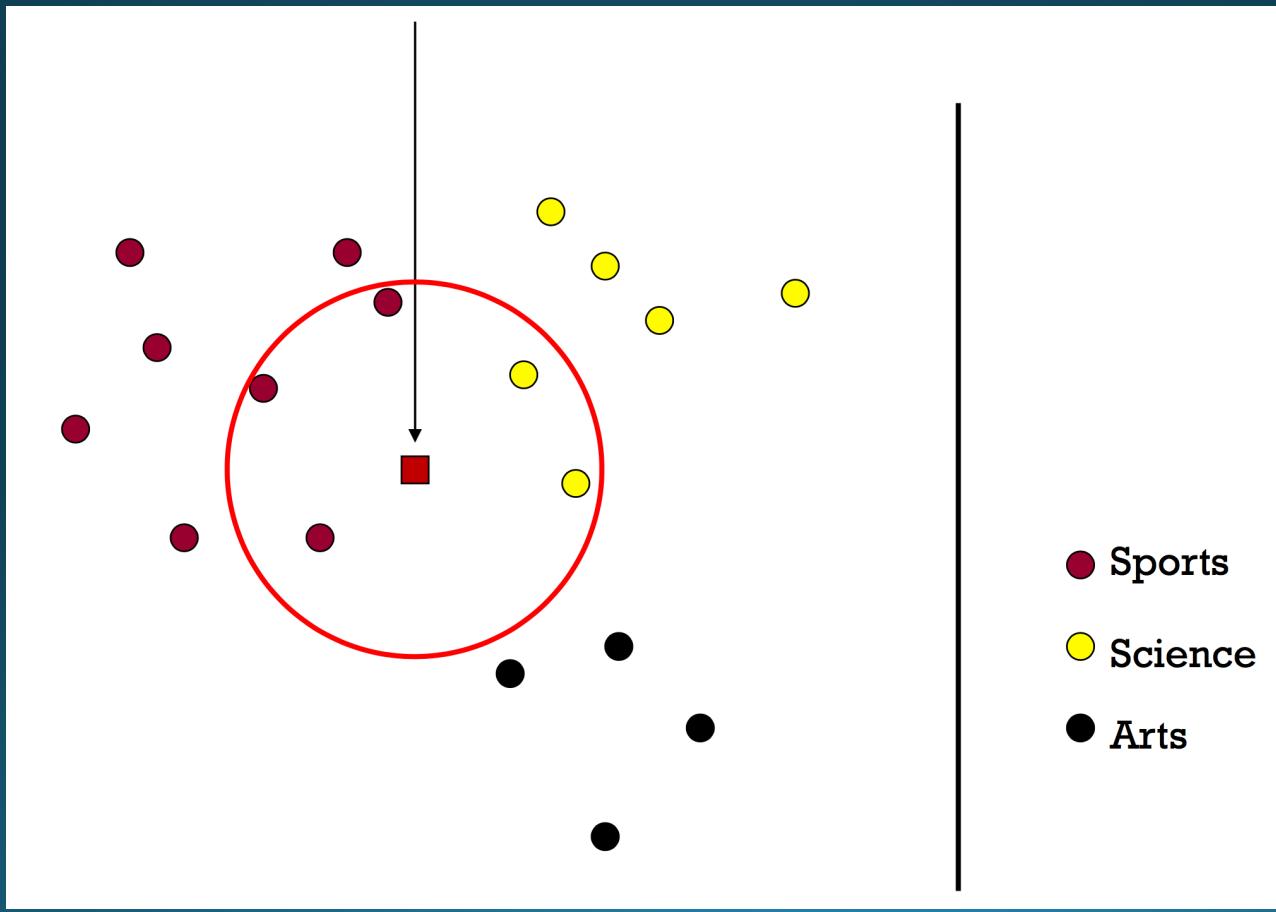
# 2-NN



# 3-NN



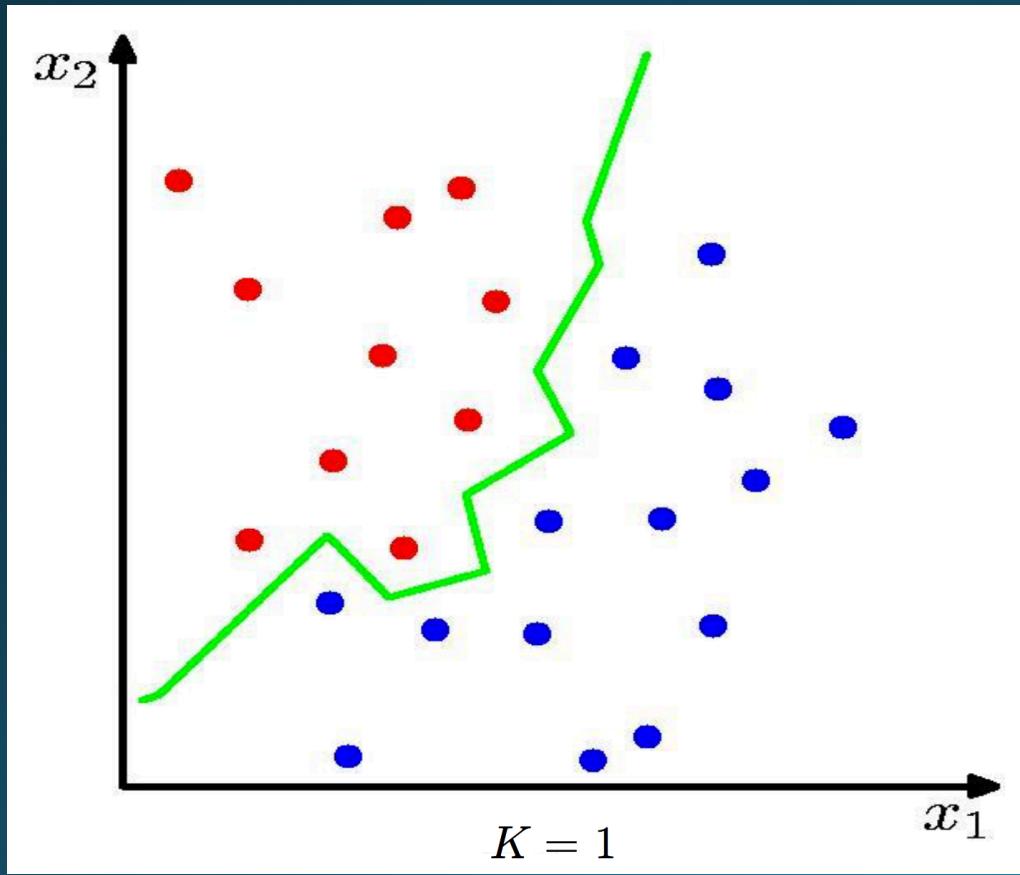
# 5-NN



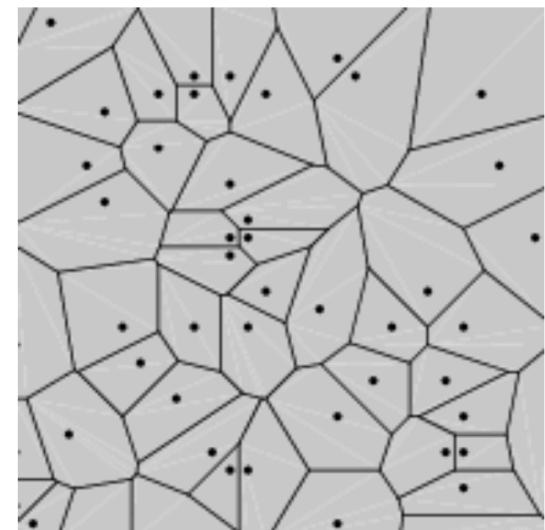
# What is the best $k$ ?

- Bias-variance trade-off
- **Large  $k$**  = predicted label is more stable
- **Small  $k$**  = predicted label is more accurate
- **Similar to density estimation**

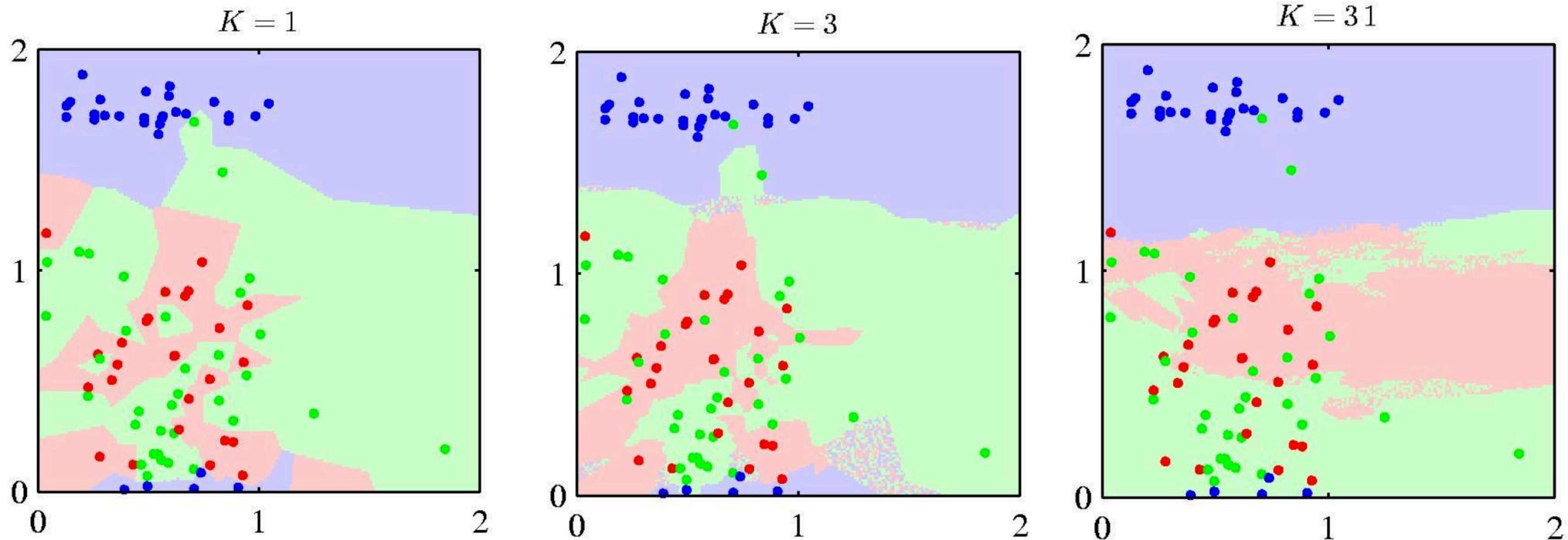
# 1-NN Decision Boundary



Voronoi  
Diagram



# KNN Decision Boundaries



- **Guarantee:** For  $n \rightarrow \infty$ , error rate of 1-NN is never more than 2x optimal error rate

# Case Study: Newsgroups Classification

- 20 Newsgroups
- 61,118 words
- 18,774 documents
- Class label 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

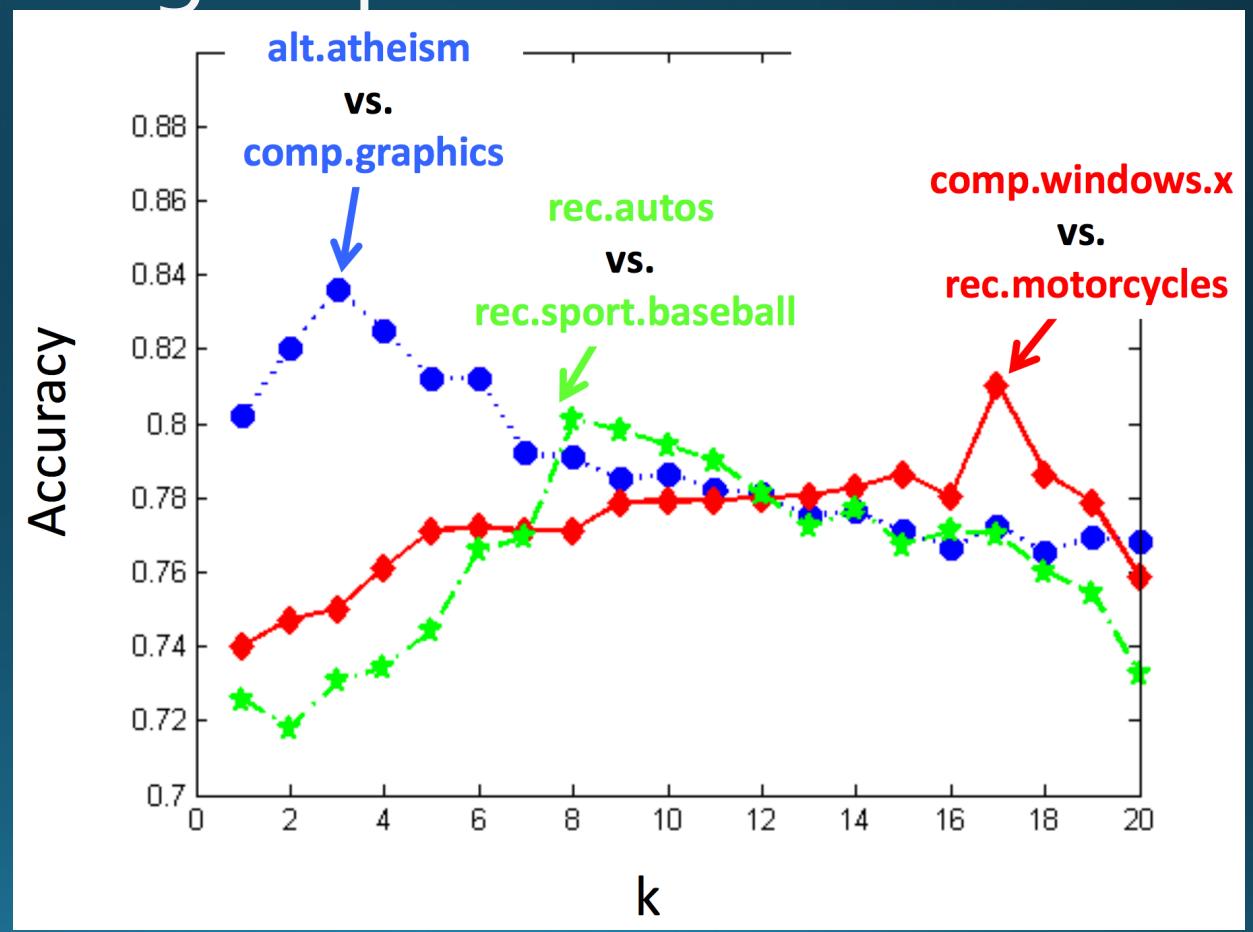
# Case Study: Newsgroups Classification

- Training/Testing
  - 50%-50% randomly split
  - 10 runs
  - Report average results
- Evaluation Criteria

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

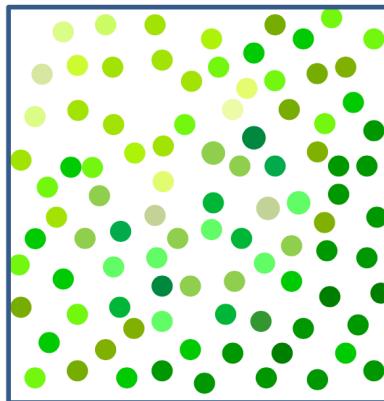
# Case Study: Newsgroups Classification

- Results in binary class comparisons



# Temperature Sensing

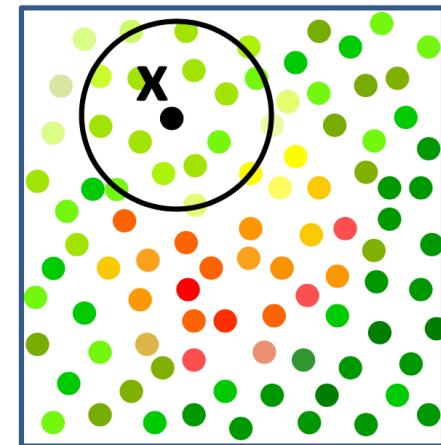
- What is the temperature in the room?



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

Average

at location  $x$ ?



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

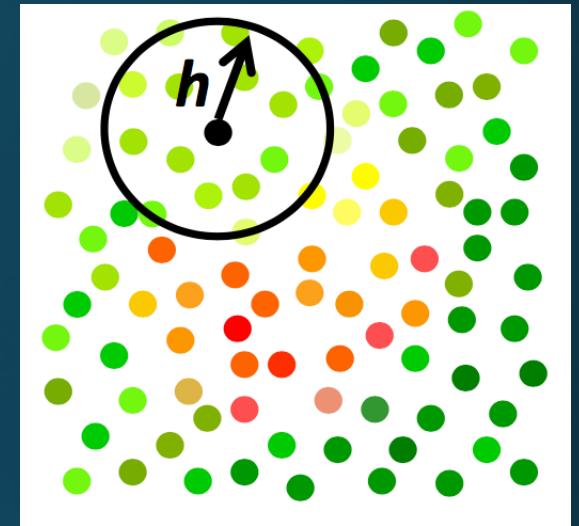
"Local" Average

# Kernel Regression

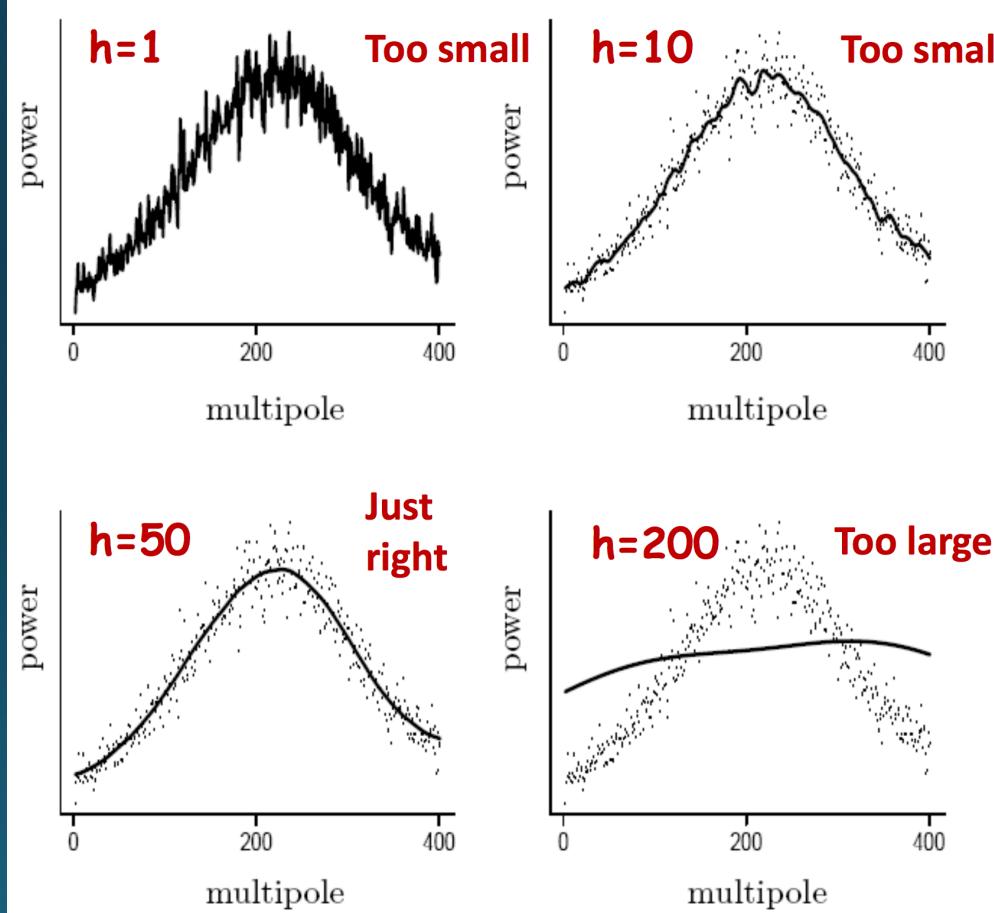
- Or “local” regression
- Nadaraya-Watson Kernel Estimator

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

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



# Choice of kernel bandwidth



Choice of *kernel* is not terribly important!

# Kernel Regression as WLS

- Weighted Least Squares (WLS) has the form

$$\min_f \sum_{i=1}^n w_i (f(X_i) - Y_i)^2$$

- Compare to Nadaraya-Watson form

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

- Kernel regression corresponds to locally constant estimator obtained from [locally] weighted least squares
- Set  $f(X_i) = \beta$  where  $\beta$  is constant

# Kernel Regression as WLS

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

A constant value

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

Individual weights have to sum to 1

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

# Summary

- Nonparametric places mild assumptions on data; good models for complex data
  - Usually requires storing & computing with full dataset
- Parametric models rely on very strong, simplistic assumptions
  - Once fitted, they are much more efficient with storage and computation
- Effects of bin width & kernel bandwidth
  - Bias-variance trade-off
- KNN classifier
  - Non-linear decision boundaries
- Kernel regression
  - Comparison to weighted least squares

# Questions?

# Course Details

- How is Assignment 5 going?
- How are the projects going?
- Workshop 8 on Autoencoders tomorrow!

# References

- “All of Nonparametric Statistics”,  
<http://www.stat.cmu.edu/~larry/all-of-nonpar/index.html>