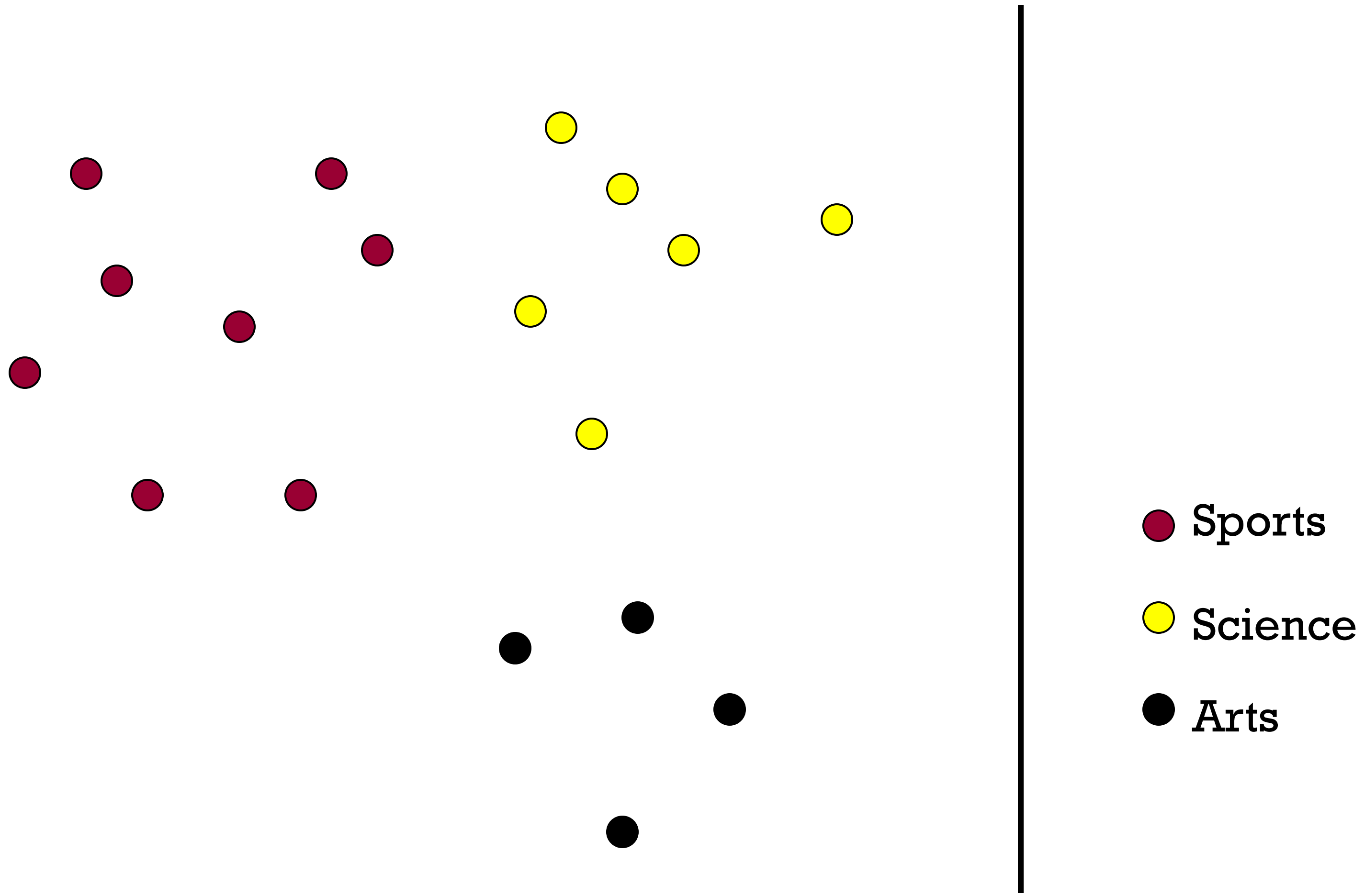
k-NN classifier
Nonparametric kernel regression

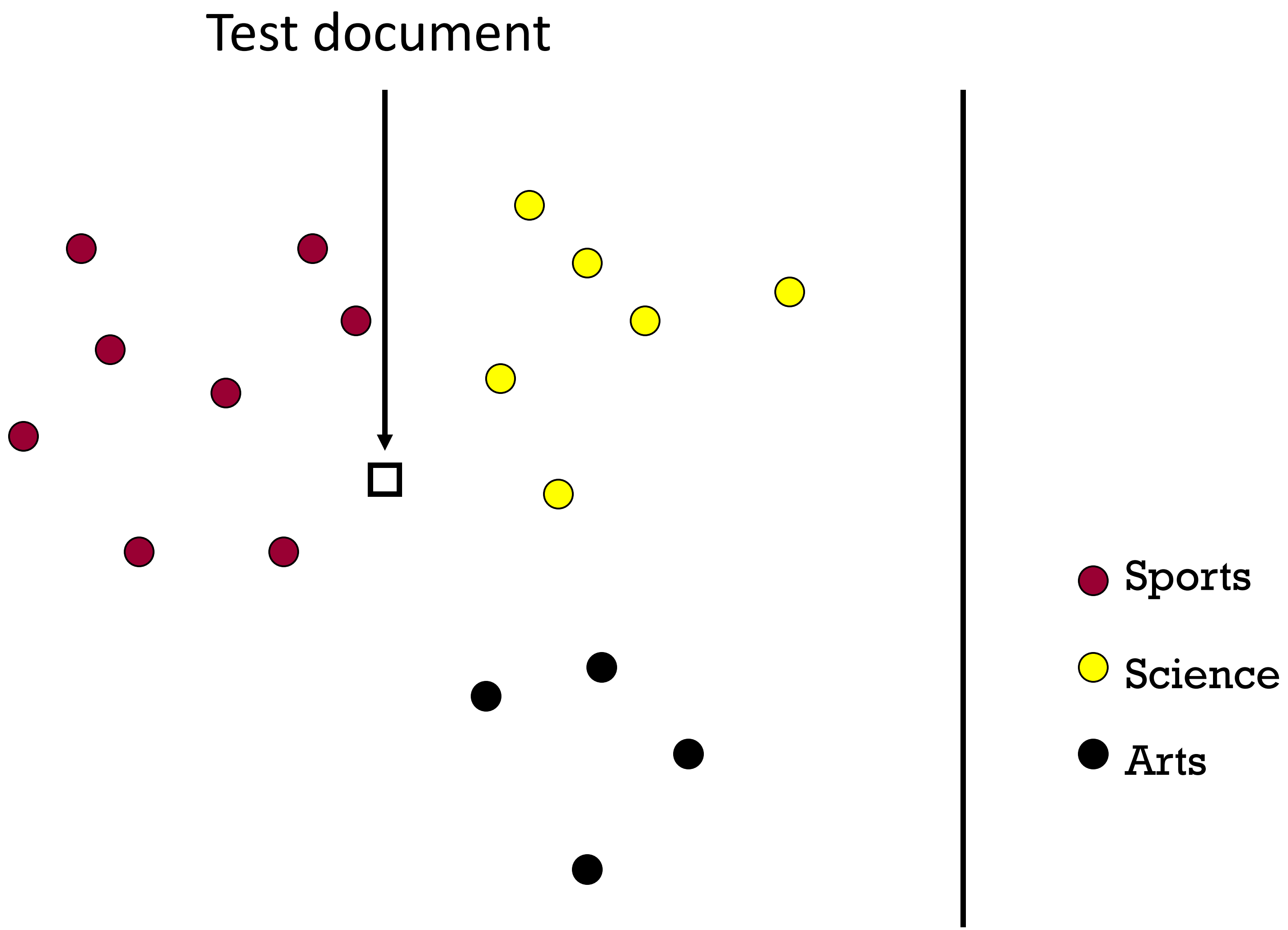
Aarti Singh

Feb 27, 2023
Machine Learning 10-701

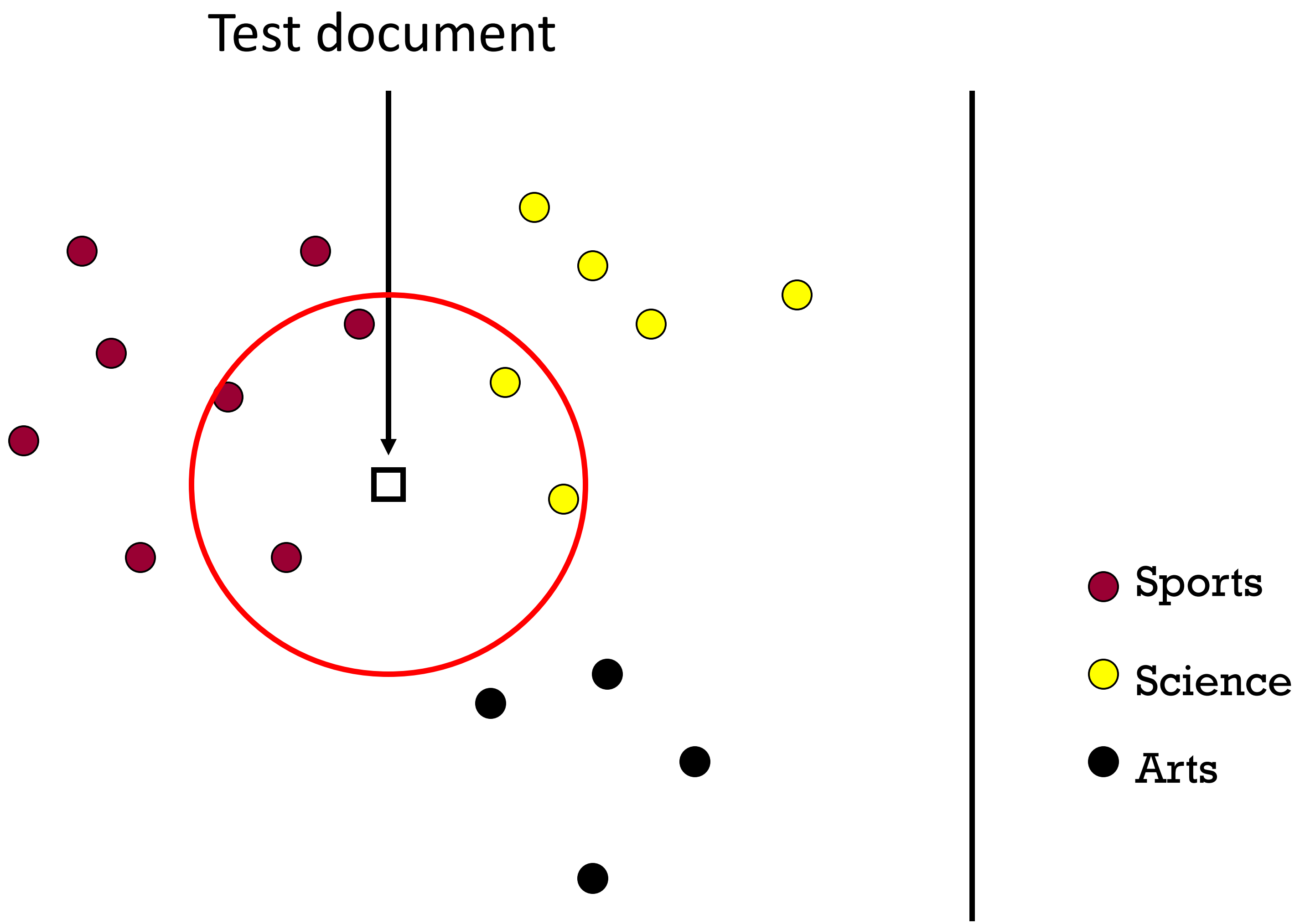
k-NN classifier



k-NN classifier



k-NN classifier (k=5)



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

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

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

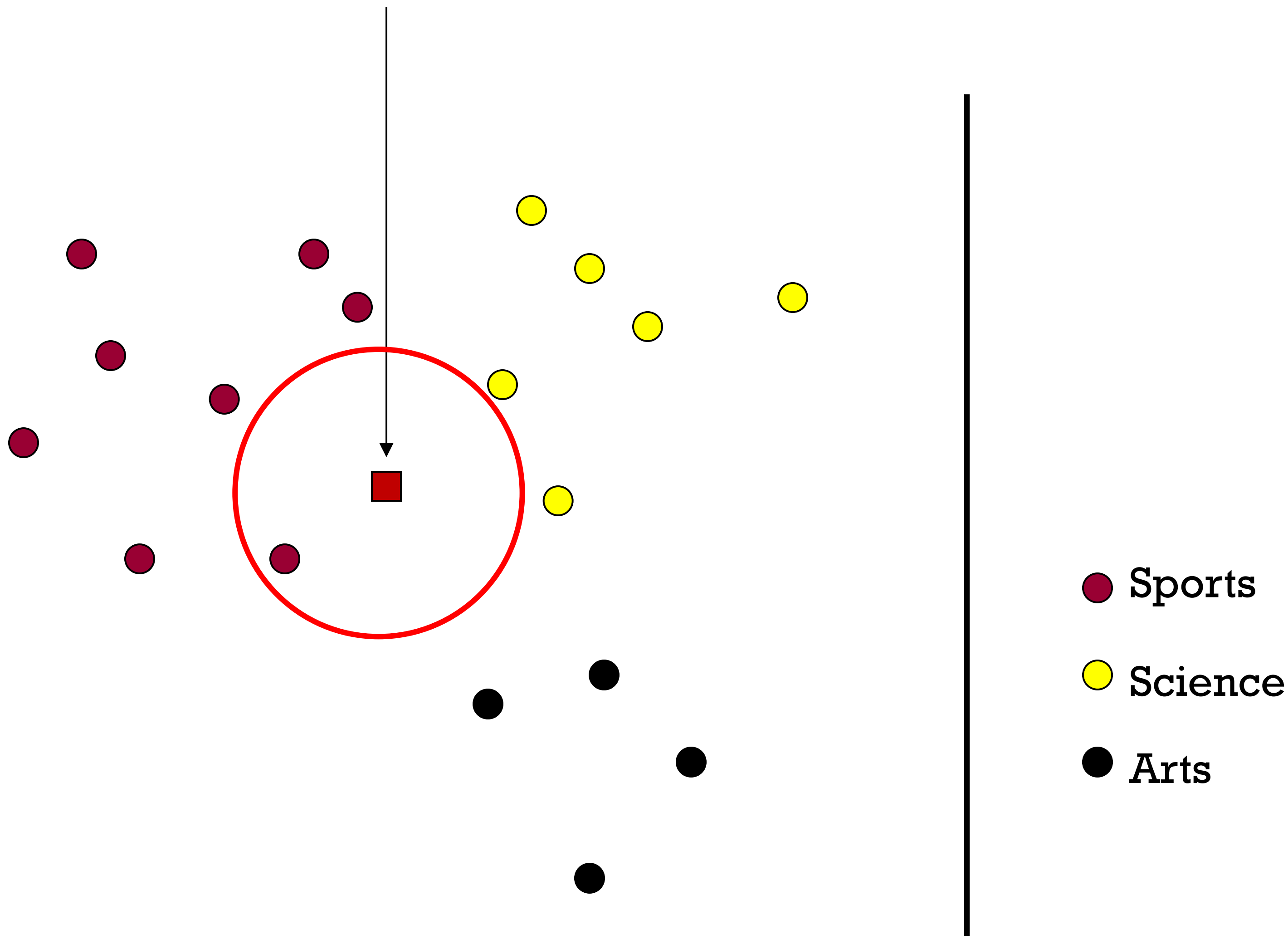
→ # training pts of class y amongst k NNs of x

→ # total training pts of class y

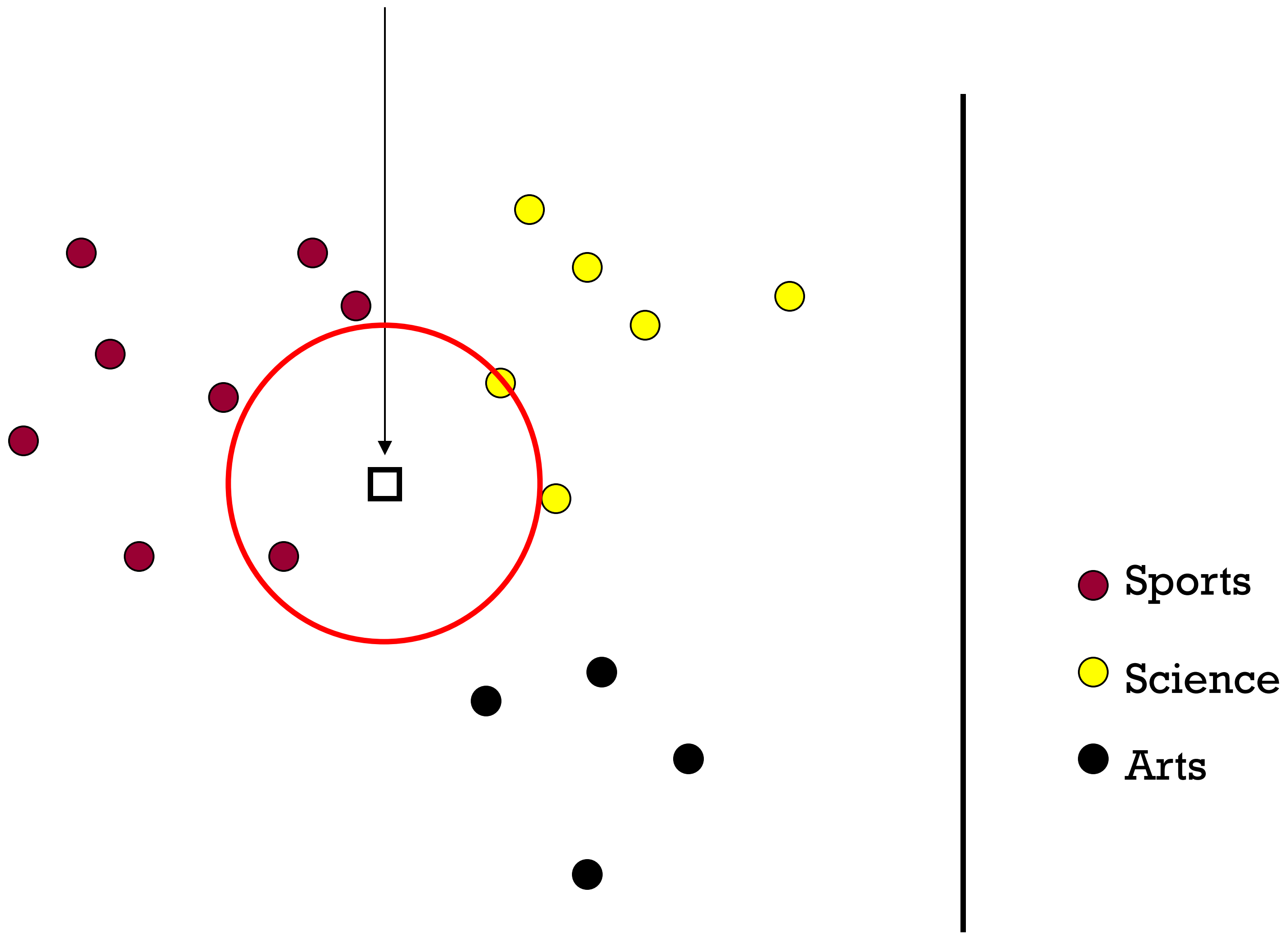
$$\sum_y k_y = k$$

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

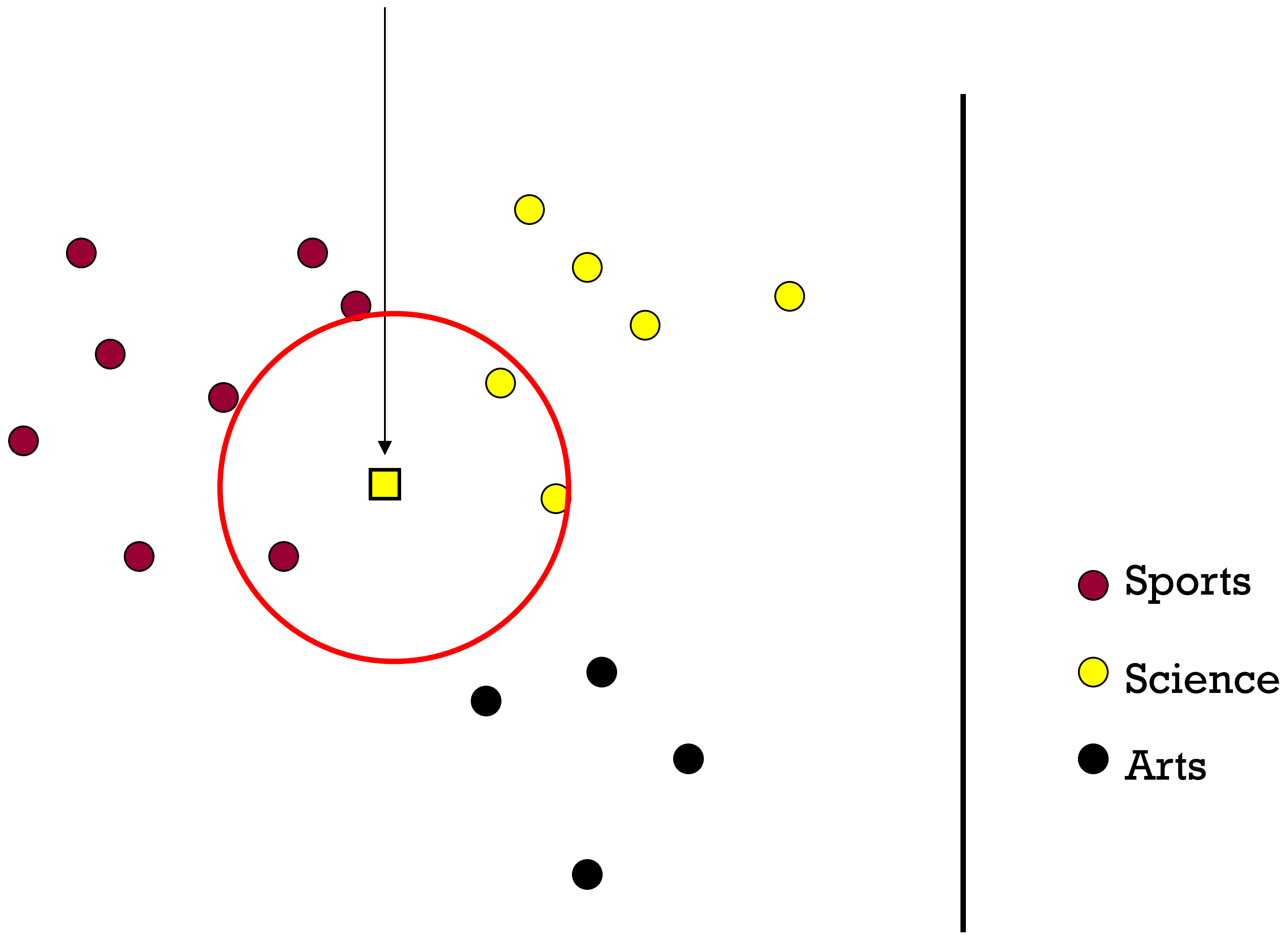
1-Nearest Neighbor (kNN) classifier



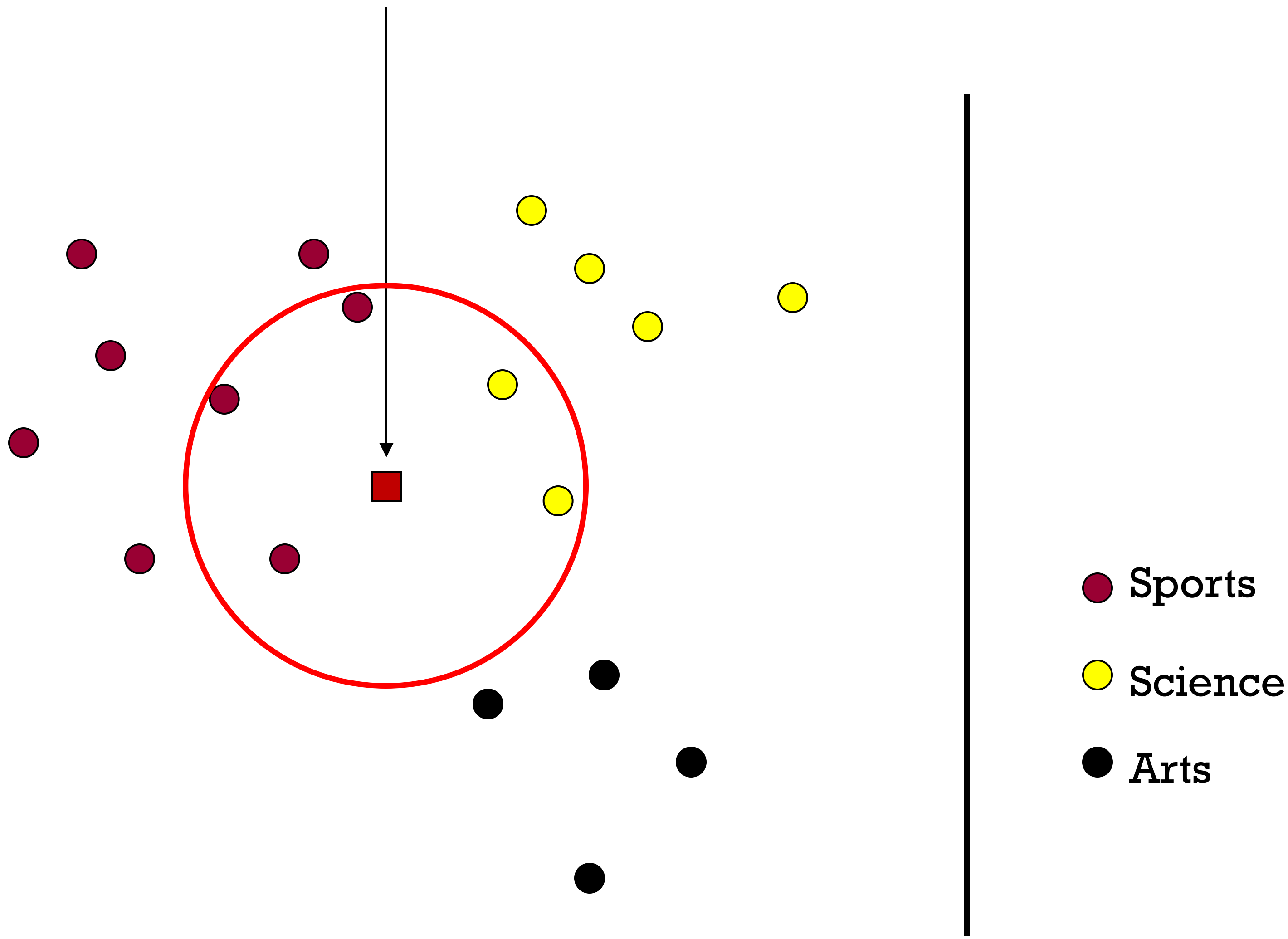
2-Nearest Neighbor (kNN) classifier



3-Nearest Neighbor (kNN) classifier

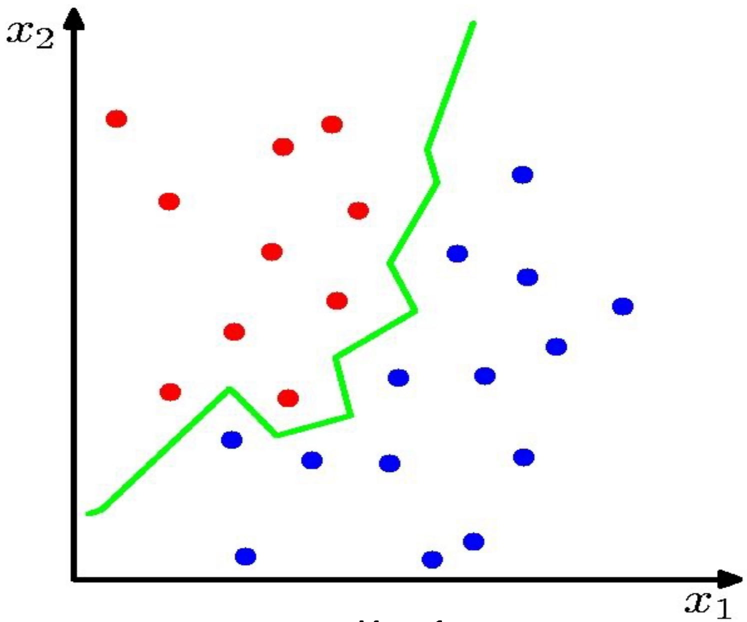


5-Nearest Neighbor (kNN) classifier



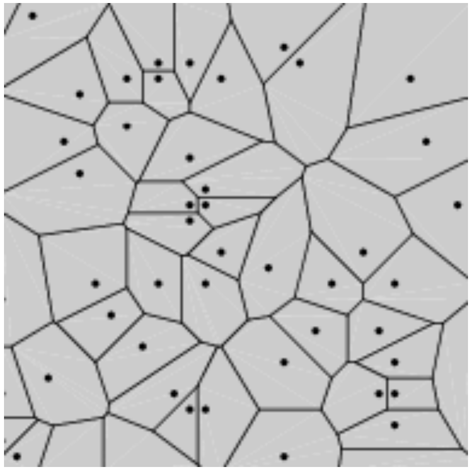
What is the best k?

1-NN classifier decision boundary



K = 1

Voronoi Diagram



As k increases, boundary becomes smoother (less jagged).

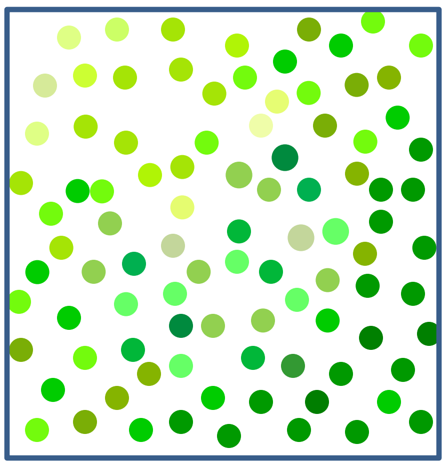
What is the best k?

Approximation vs. Stability (aka Bias vs Variance) Tradeoff

- Larger K => predicted label is more stable (low variance) but potentially less accurate (high bias)
- Smaller K => predicted label can approximate best classifier well given enough data (low bias) but predict label is unstable (high variance)

Local Kernel Regression

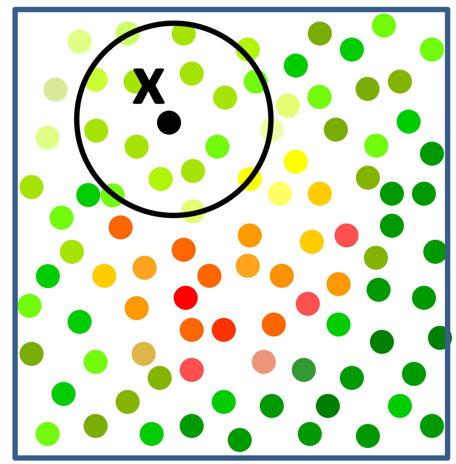
- What is the temperature in the room?



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

Average

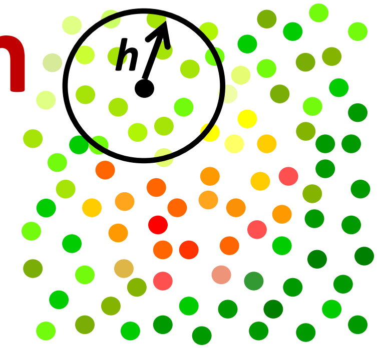
at location x?



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

"Local" Average

Local Kernel Regression



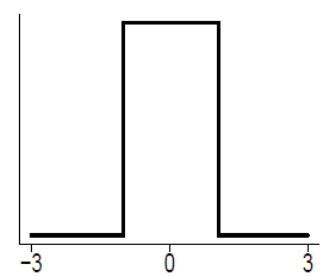
- Nonparametric estimator
- Nadaraya-Watson Kernel Estimator

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

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

boxcar kernel :

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



Choice of kernel bandwidth h

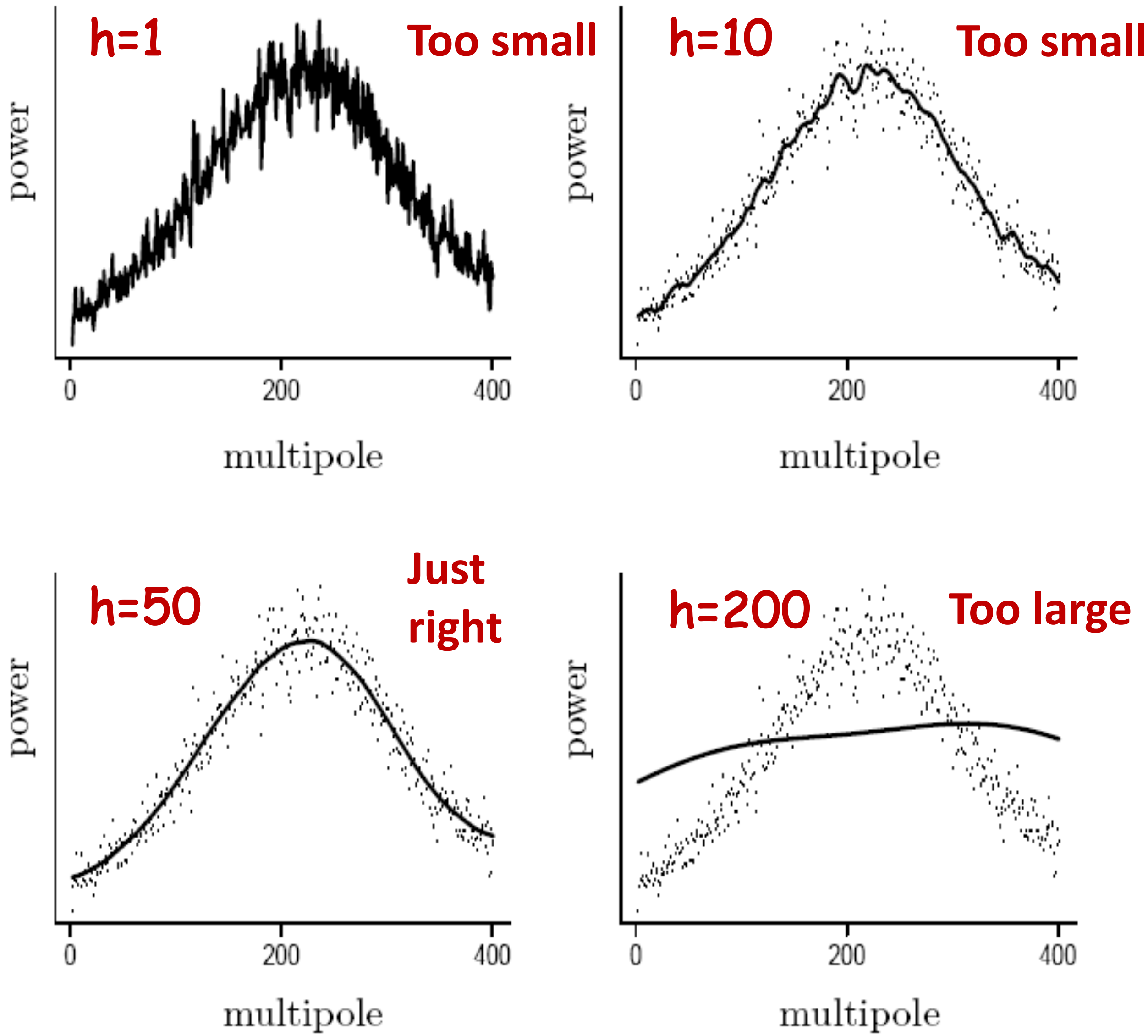


Image Source: Larry's book – All of Nonparametric Statistics

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)

$$\min_{\beta} \sum_{i=1}^{n} w_i(\underset{\text{constant}}{\underset{\downarrow}{\beta}} - Y_i)^2$$

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

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

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) modeling assumptions

  - Nonparametric models typically require storage and computation of the order of entire data set size.

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