Non-parametric methods

Aarti Singh

Machine Learning 10-315
Oct 11, 2021

Parametric methods

- Assume some model (Gaussian, Bernoulli, Multinomial, logistic, network of logistic units, Linear, Quadratic) with fixed number of parameters
  - Gaussian Bayes, Naïve Bayes, Logistic Regression, Neural Networks

- Estimate parameters ($\mu, \sigma^2, \theta, w, \beta$) using MLE/MAP and plug in

- **Pro** – need few data points to learn parameters
- **Con** – Strong distributional assumptions, not satisfied in practice

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

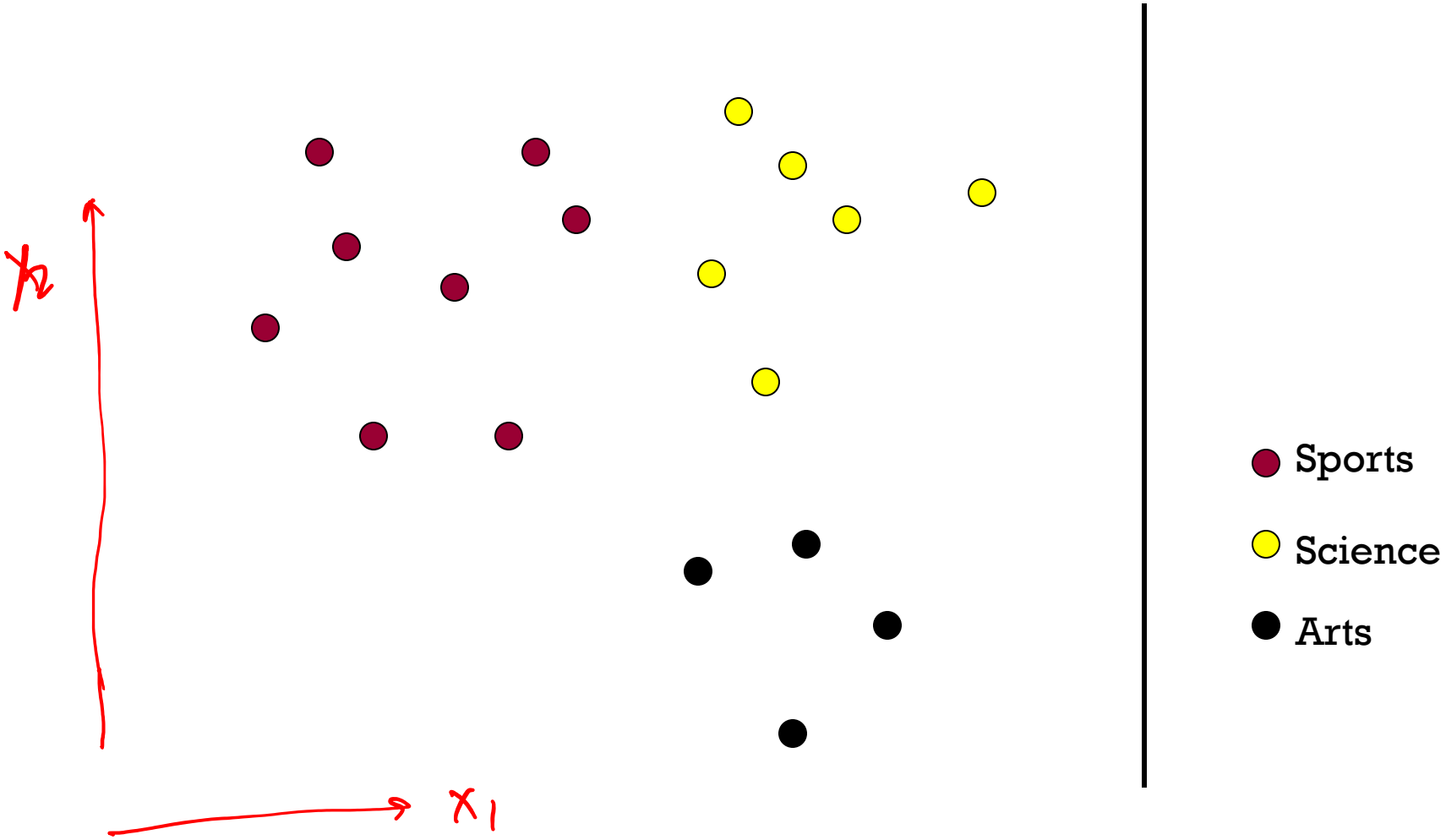
- Some nonparametric methods

  Classification: Decision trees, k-NN (k-Nearest Neighbor) classifier

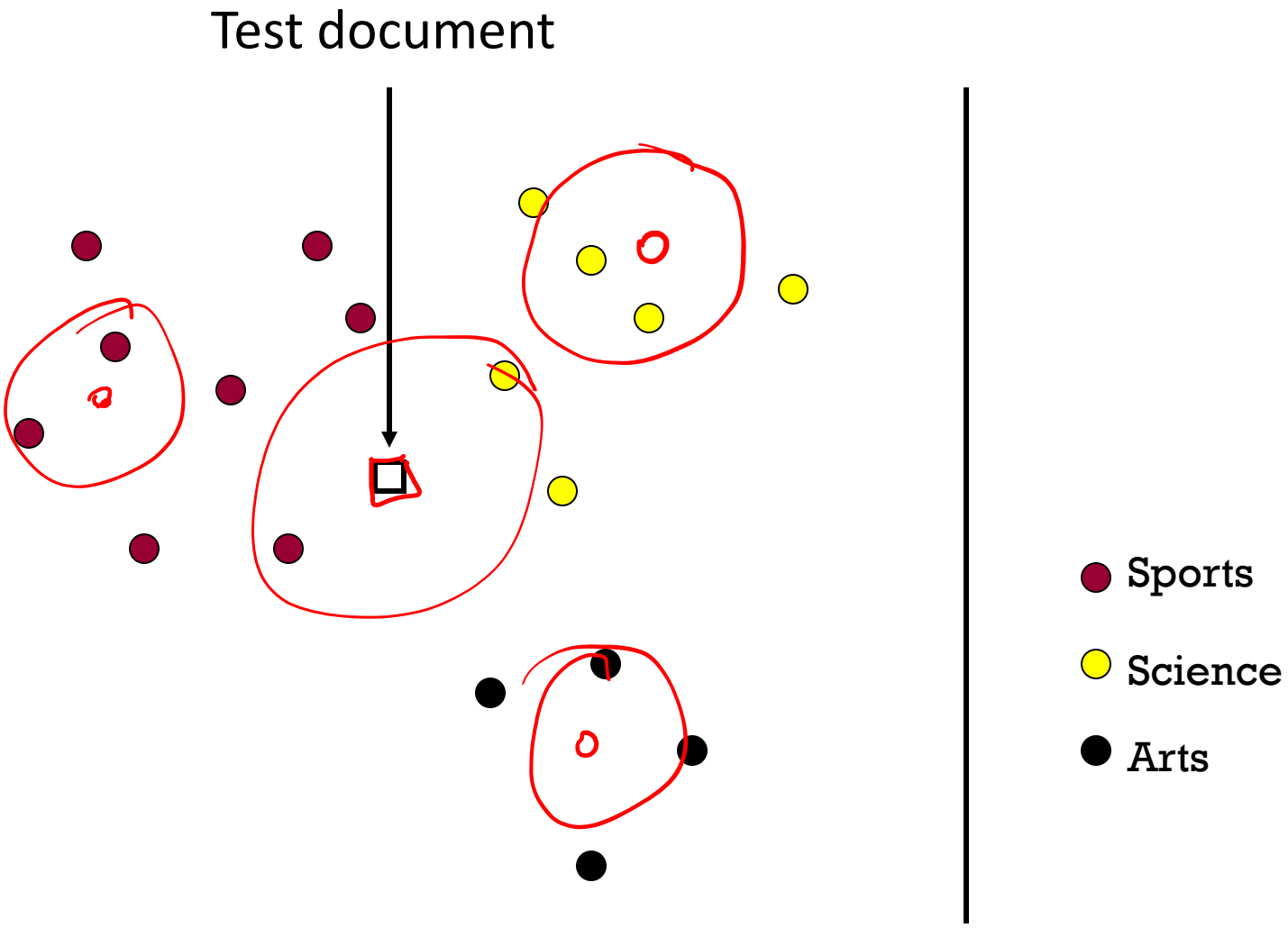
  Density estimation: k-NN, Histogram, Kernel density estimate

  Regression: Kernel regression

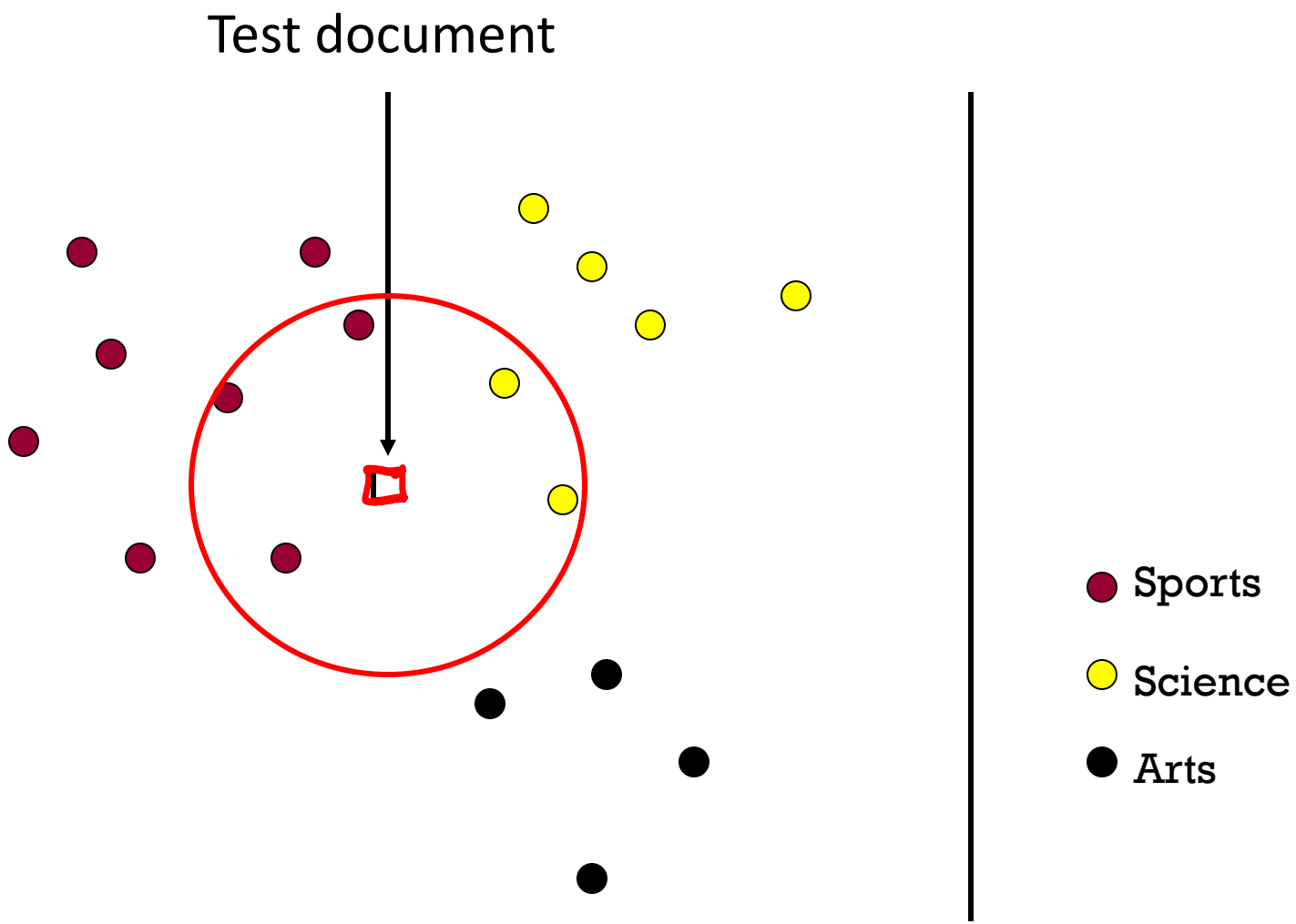
k-NN classifier



k-NN classifier



k-NN classifier (k=5)



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

k-NN classifier

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

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

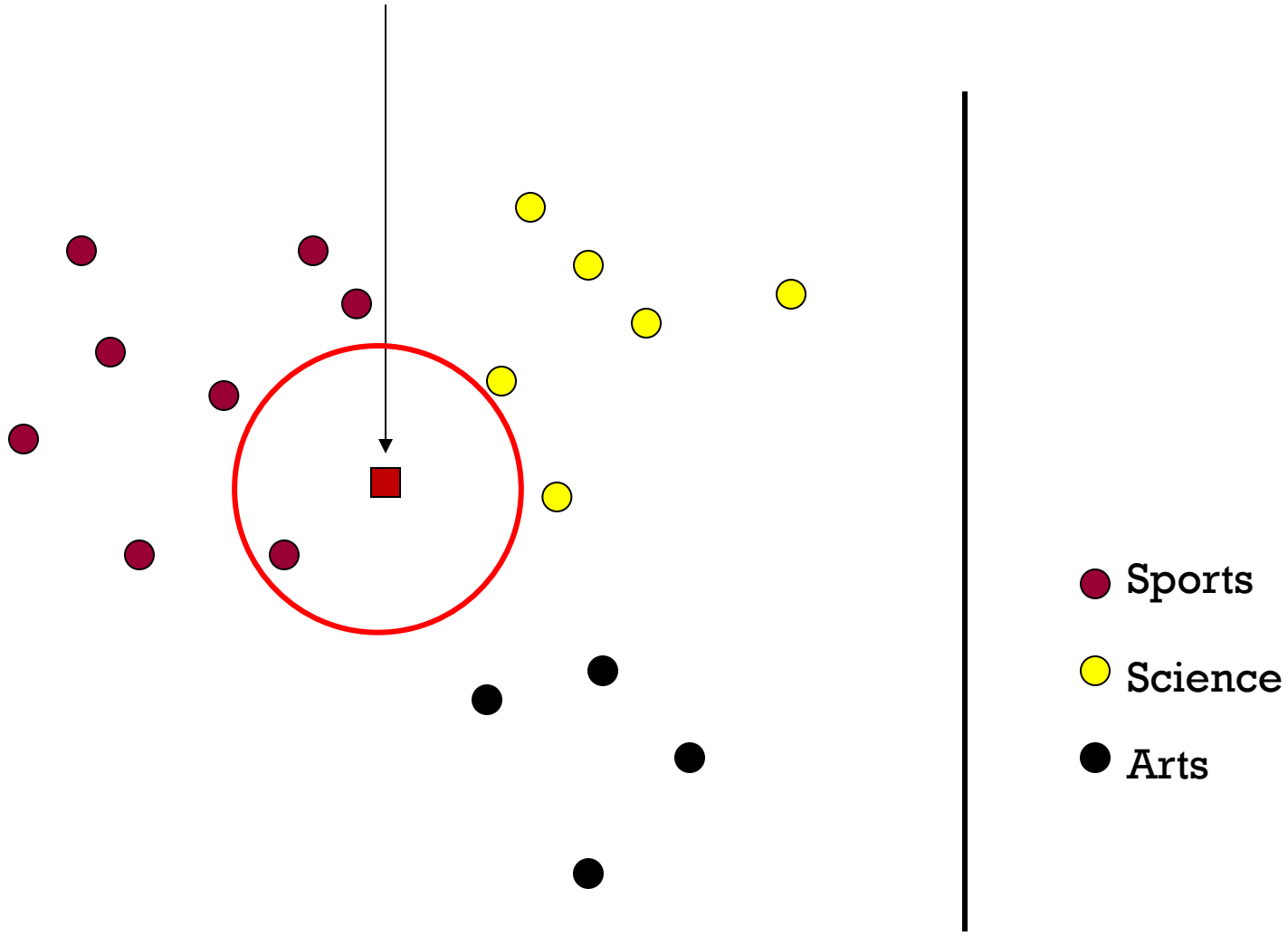
$\hat{P}_{kNN}(x|y) = \frac{k_y}{n_y}$ → # training pts of class y amongst k NNs of x

→ # total training pts of class y

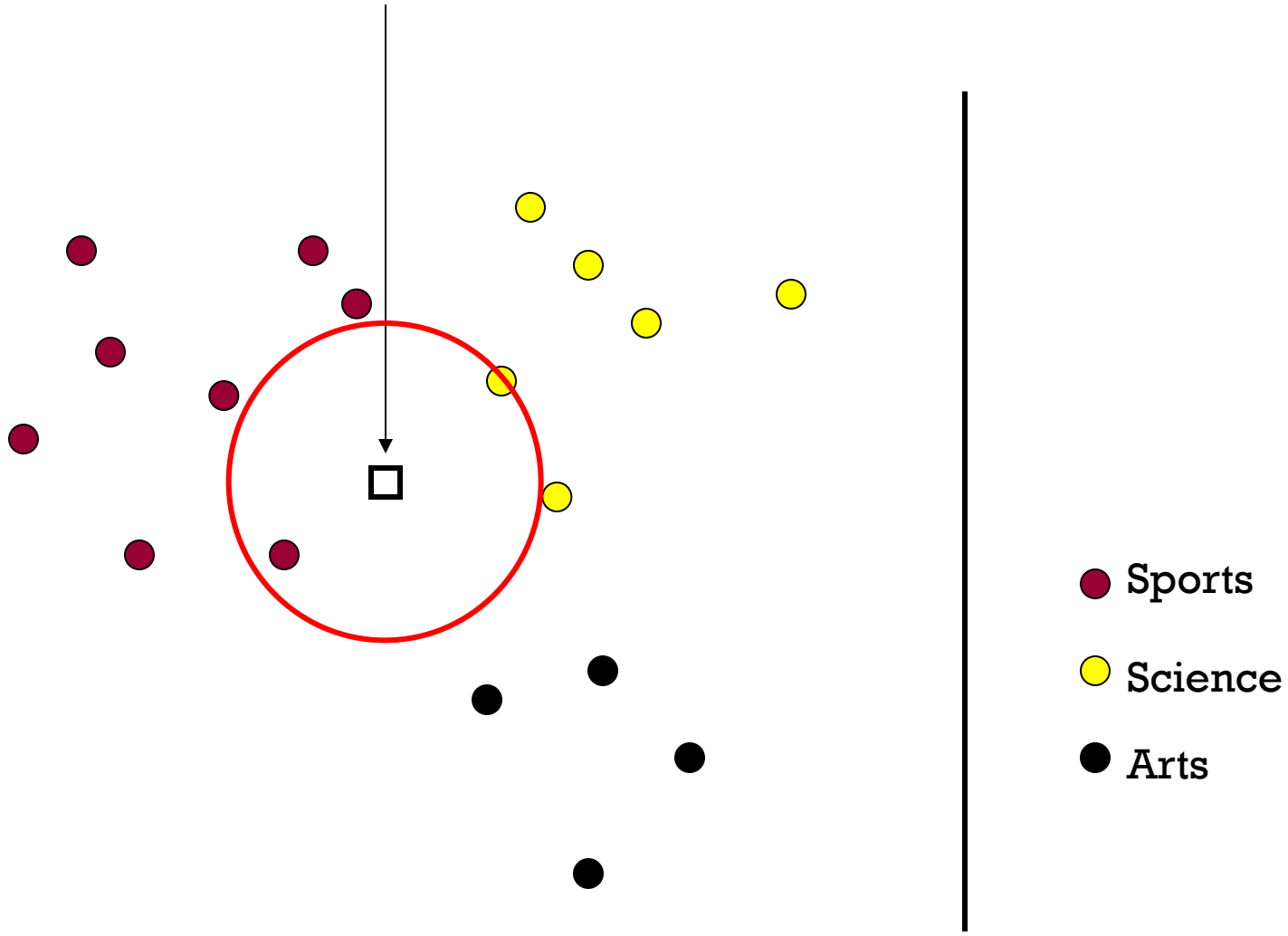
$\sum_y k_y = k$

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

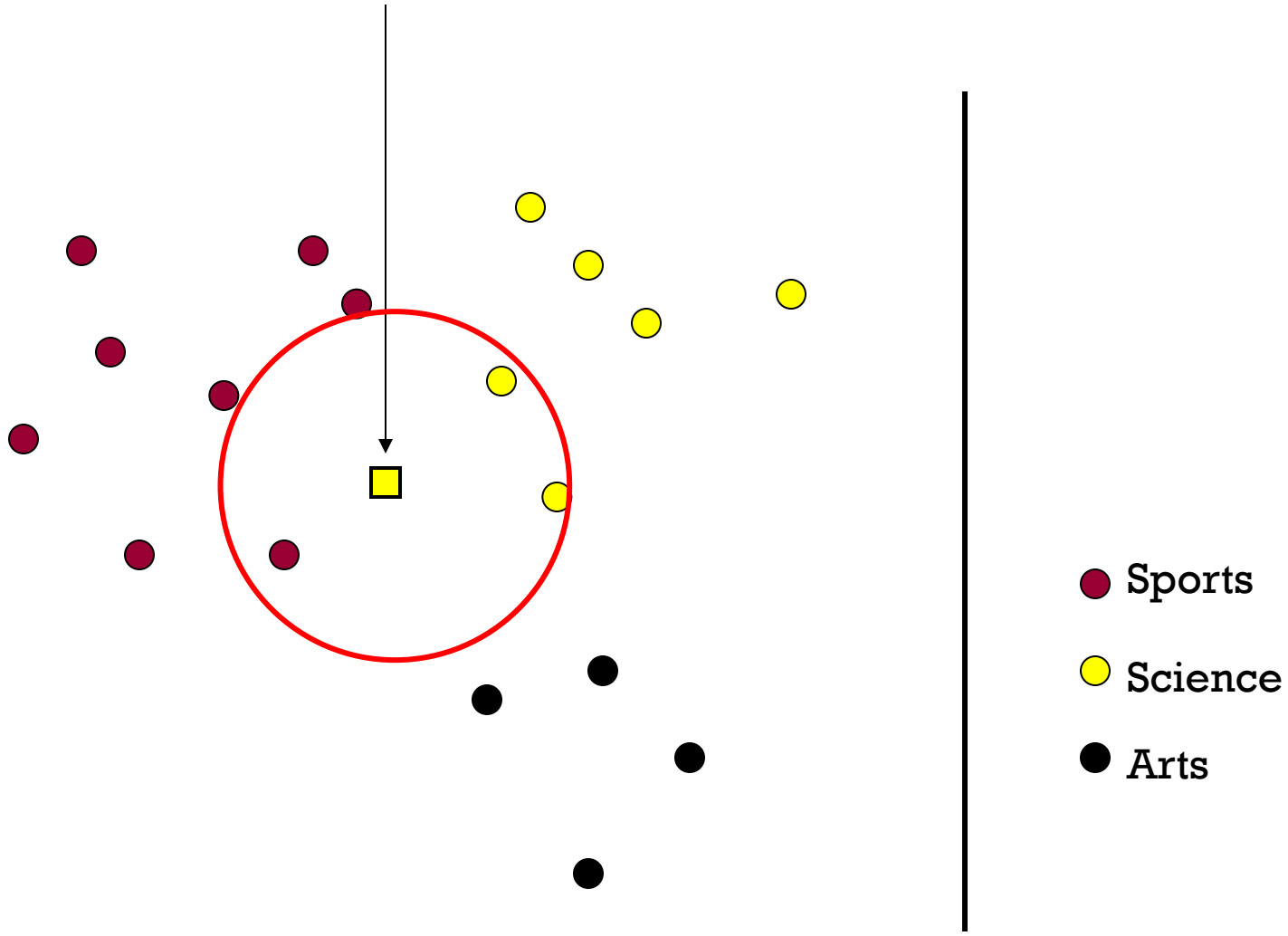
1-Nearest Neighbor (kNN) classifier



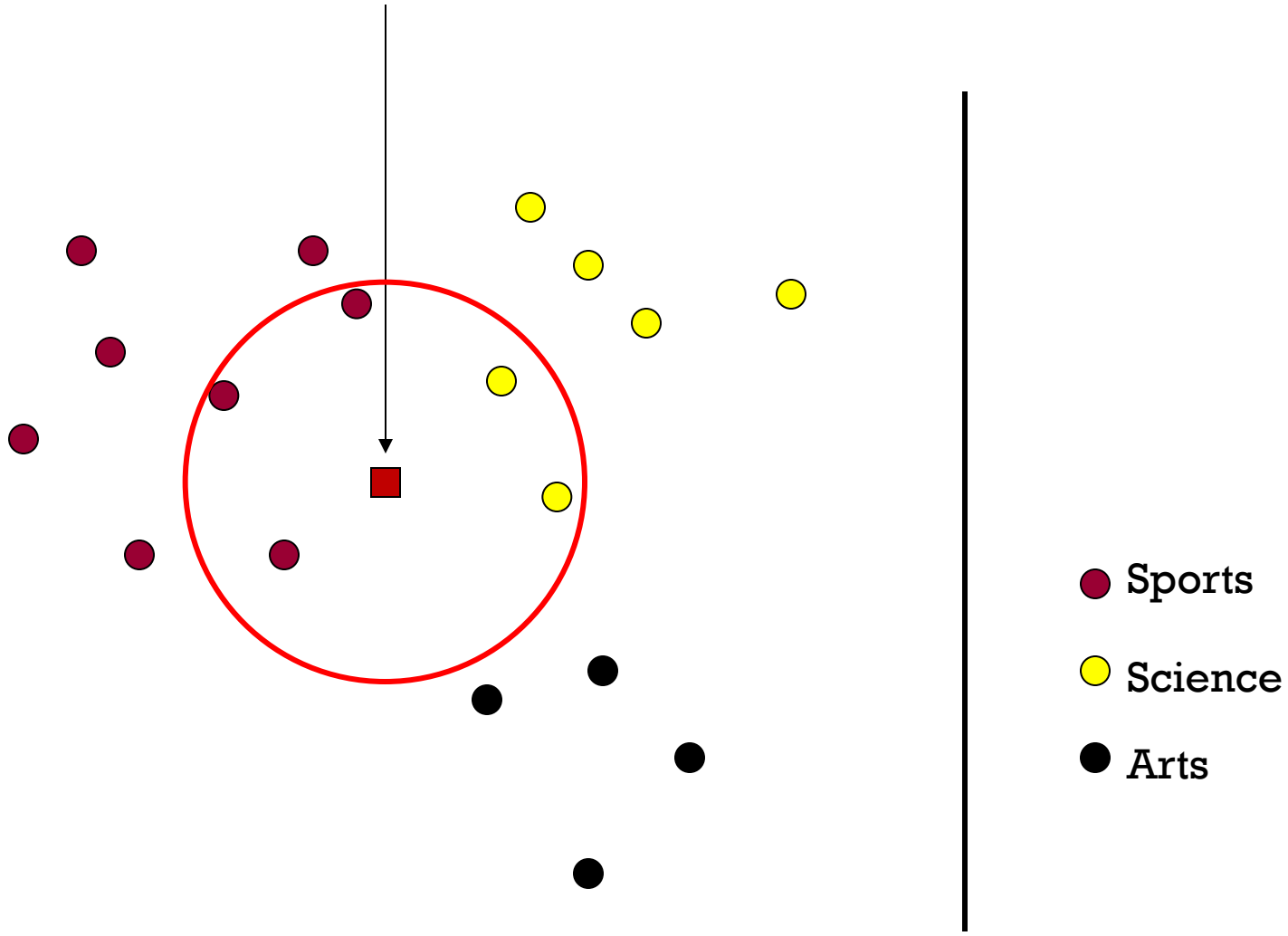
2-Nearest Neighbor (kNN) classifier



3-Nearest Neighbor (kNN) classifier

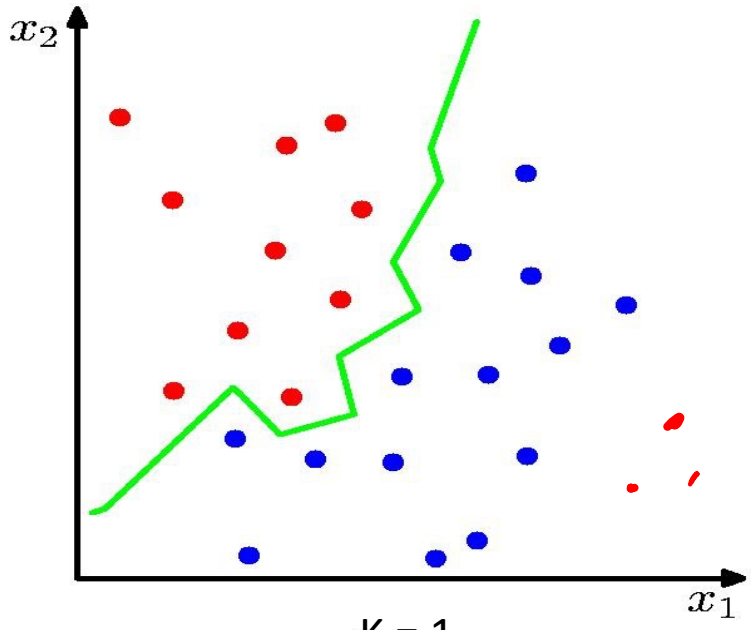


5-Nearest Neighbor (kNN) classifier



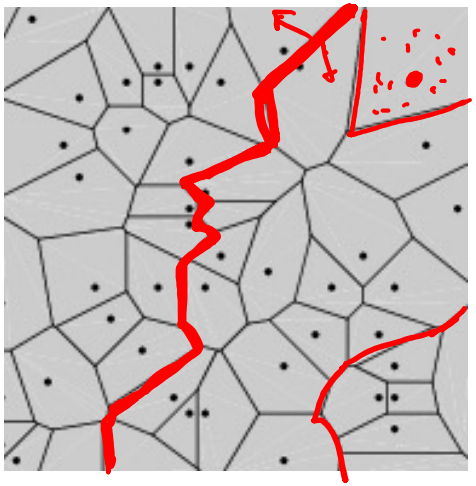
What is the best k?

1-NN classifier decision boundary



K = 1

Voronoi Diagram



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

What is the best k?

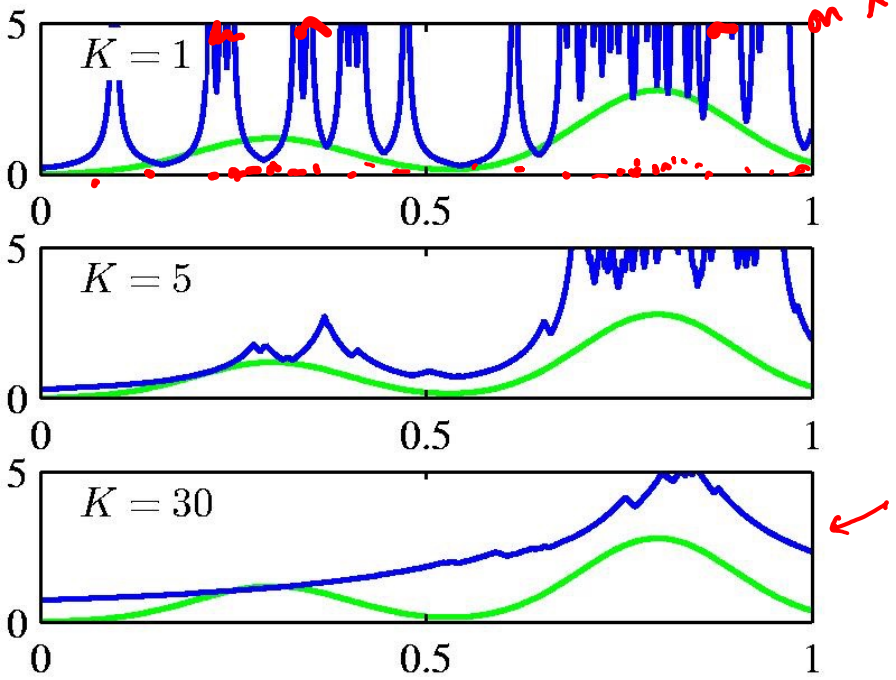
Approximation vs. Stability (aka Bias vs Variance) Tradeoff

- Larger K => predicted label is more stable
- Smaller K => predicted label can approximate best classifier well given enough data

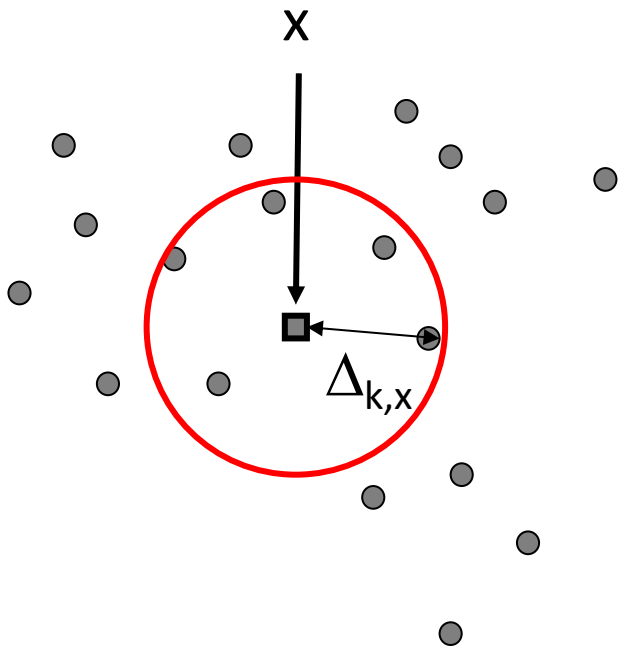
k-NN density estimation

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

volume of k-NN ball centered on 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.

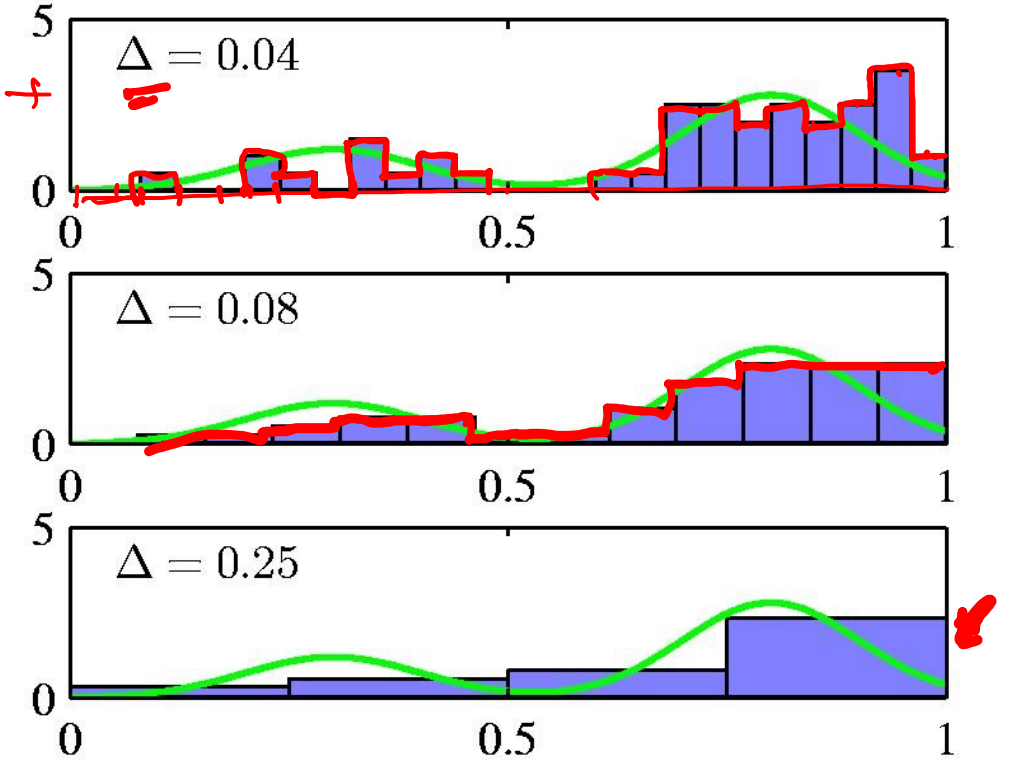


Image src: Bishop book

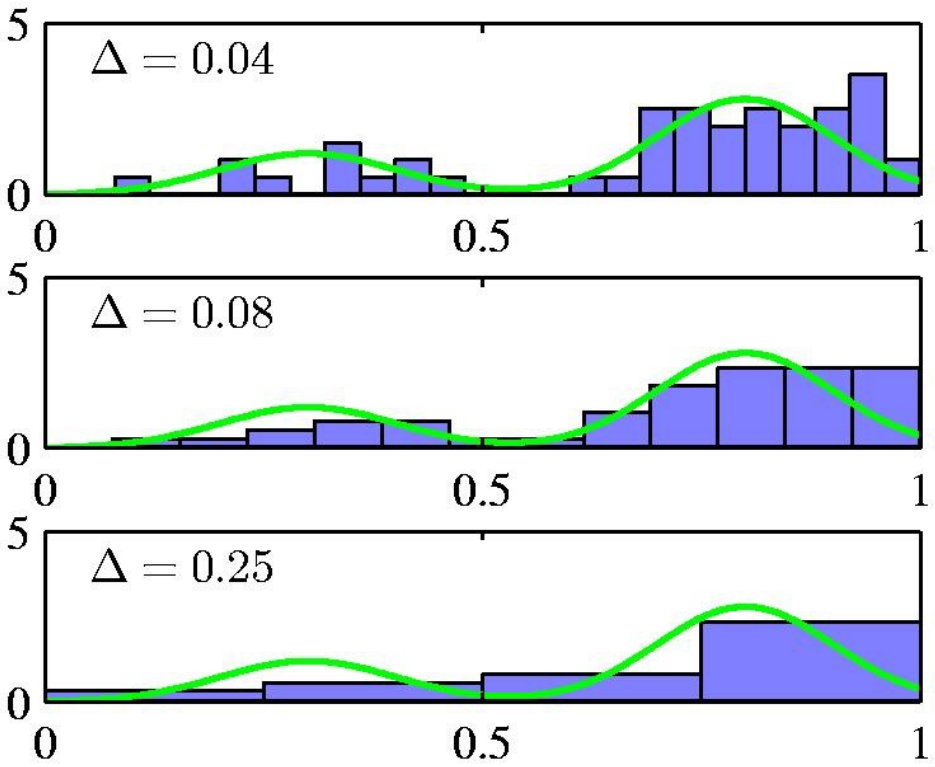
Effect of histogram bin width

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

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**"

bins = $1/\Delta$



Histogram as MLE

- Underlying model – density is constant on each bin

  Parameters p_j : density in bin j

  $P(X \in Bin_j) = p_j \Delta$

  Note $\sum_j p_j = 1/\Delta$ since $\int p(x)dx = 1$

- Maximize likelihood of data under probability model with parameters p_j

  $P(D|\theta)$

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

- Show that histogram density estimate is MLE under this model

Histogram as MLE

$\hat{p}_{j,MLE} = \frac{n_j}{n\Delta}$

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

likelihood

A. $\prod_{j=1}^{1/\Delta} p_j^{n_j}$ *where* n_j *– number of data in bin* j

B. $\prod_{j=1}^{n} p_j$

C. $\prod_{j=1}^{n} p_j^{1/\Delta}$

$$\prod_{i=1}^{n} p_j \mathbb{1}_{X_i \in \text{Bin } j} = \prod_{j=1}^{1/\Delta} p_j^{n_j}$$

$$\arg\max_{\{p_j\}} \log\left(\prod_{j=1}^{1/\Delta} p_j^{n_j}\right) = \sum_{j=1}^{1/\Delta} n_j \log p_j \quad \text{s.t.} \quad \sum_j p_j = 1/\Delta$$

$$\sum_{j=1}^{1/\Delta} n_j \log p_j - \alpha\left(\sum_j p_j - 1/\Delta\right)$$

Lagrange multiplier

$$\frac{\partial}{\partial p_j} = \frac{n_j}{p_j} - \alpha = 0$$

$$\Rightarrow \boxed{p_j = \frac{n_j}{\alpha}} \Rightarrow \sum_j p_j = \sum_j \frac{n_j}{\alpha} = \frac{n}{\alpha} = \frac{1}{\Delta} \Rightarrow \boxed{\alpha = n\Delta}$$