# STAT 480 Statistical Computing Applications

## Unit 4. Classification

# Lecture 3. $k$–Nearest Neighbors

Department of Statistics, Iowa State University

Spring 2017

# Motivation

- In theory we would like to predict qualitative responses using the Bayes classifier.
- For real data, we DO NOT know the conditional distribution of $Y$ given $X$, so computing the Bayes classifier is impossible.
- Many approaches attempt to estimate the conditional distribution of $Y$ given $X$, then classify a given observation to the class with highest estimated probability, and $k$–nearest neighbors (kNN) is one of them.

# $k$–Nearest Neighbors

1. Given a positive integer $k$ and a test observation $x_0$, the kNN classifier first identifies the $k$ points in the training data that are closest to $x_0$, represented by $N_0$.
2. It then estimates the conditional probability for class $j$ as the fraction of points in $N_0$ whose response values equal $j$

$$P(Y = j|X = x_0) = \frac{1}{K} = \sum_{i \in N_0} I(y_i = j).$$

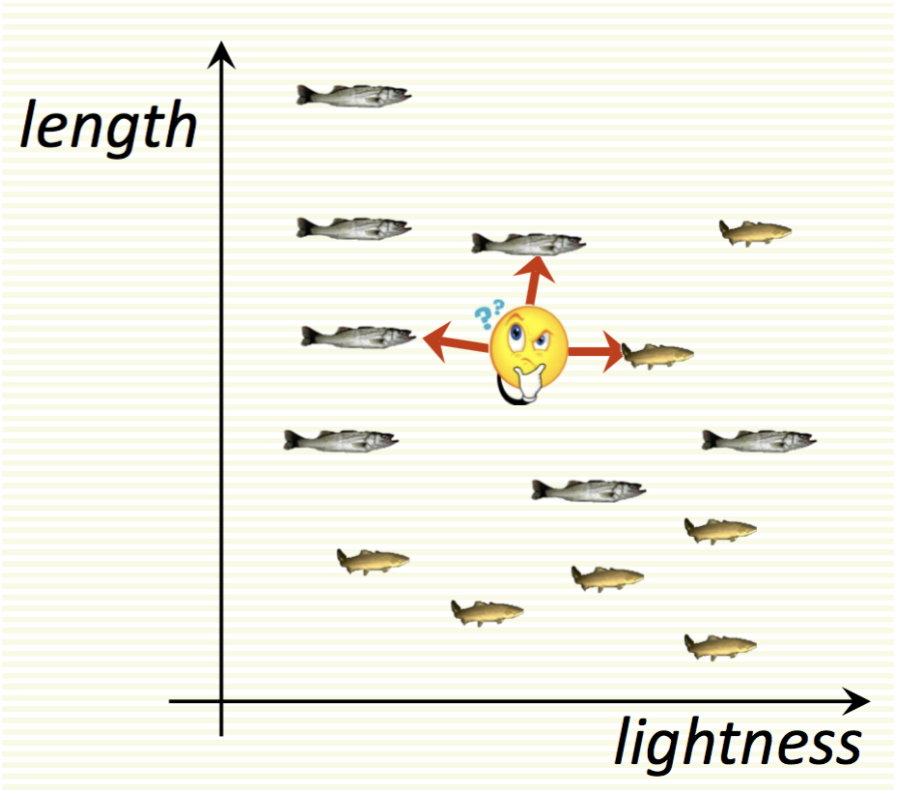
3. Finally, kNN applies Bayes rule and classifies the test observation $x_0$ to the class with the largest probability.

# $k$–Nearest Neighbors (Cont.)

- kNN classifies an unknown example with the most common class among $k$ closest examples.
  - "tell me who your neighbors are, and I'll tell you who you are"

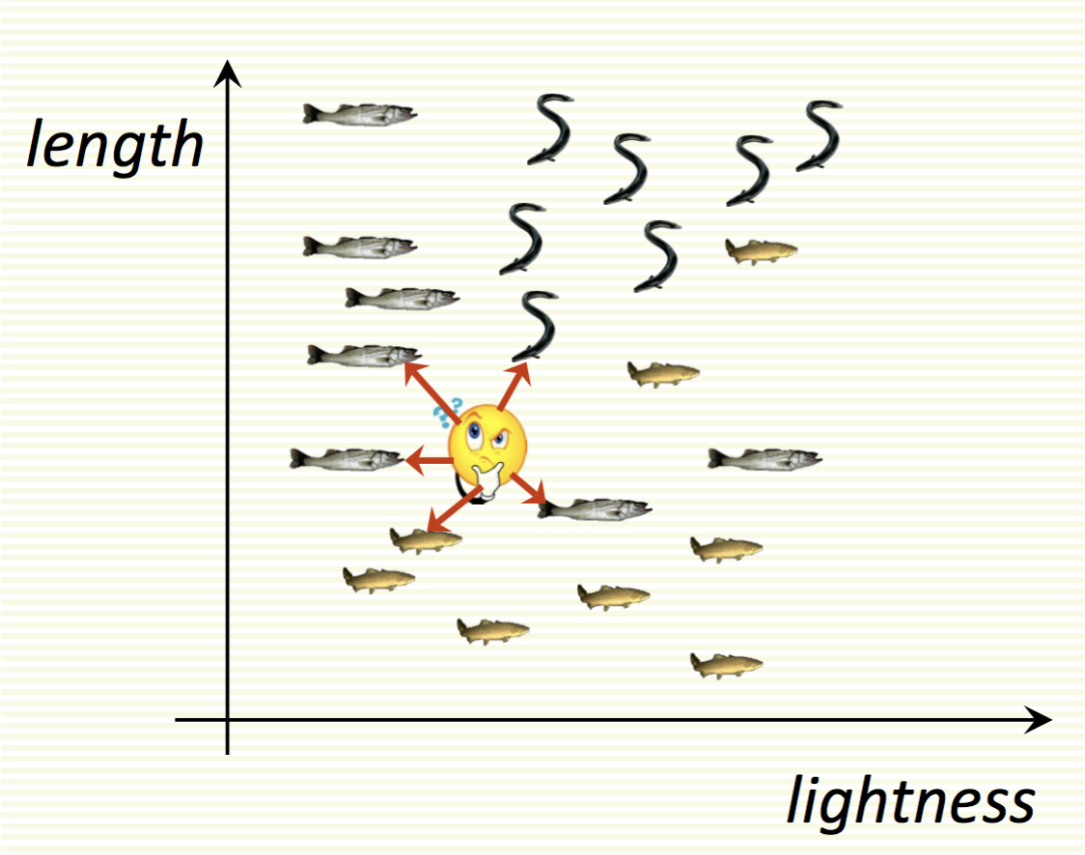
Example:

- $k = 3$
- 2 sea bass,
  1 salmon
- classify as
  sea bass



# kNN: Multiple Classes

- Easy to implement for multiple classes
- Example for $k = 5$
  - 3 fish species: sea bass, salmon, eel
  - 3 sea bass, 1 salmon, 1 eel $\Rightarrow$ classify as sea bass

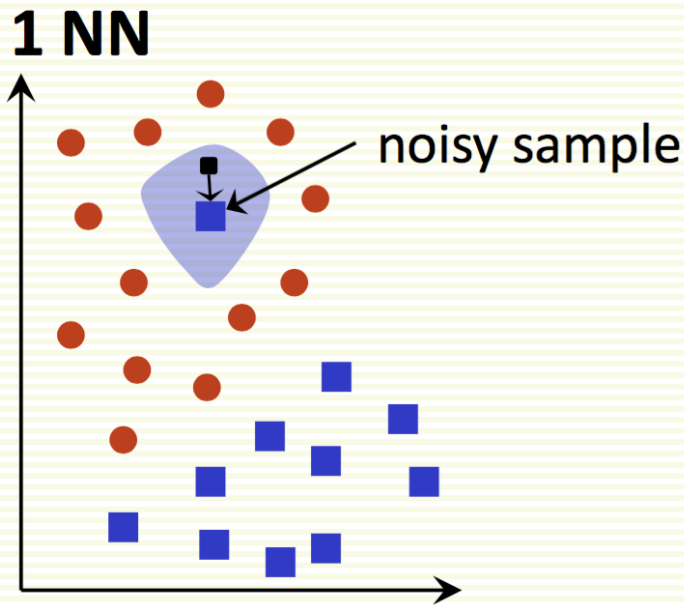


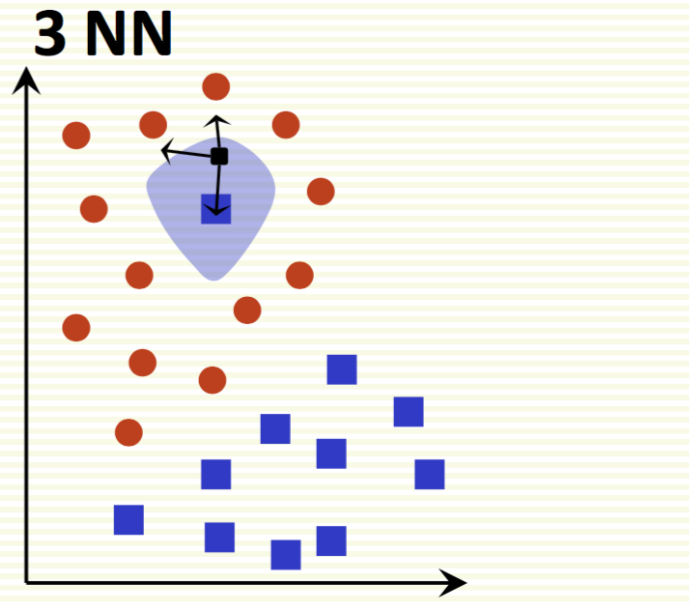
# How to Choose $k$

- In theory, if infinite number of samples available, the larger is $k$, the better is classification.
- The caveat is that all $k$ neighbors have to be close.
  - Possible when infinite # samples available
  - Impossible in practice since # samples is finite

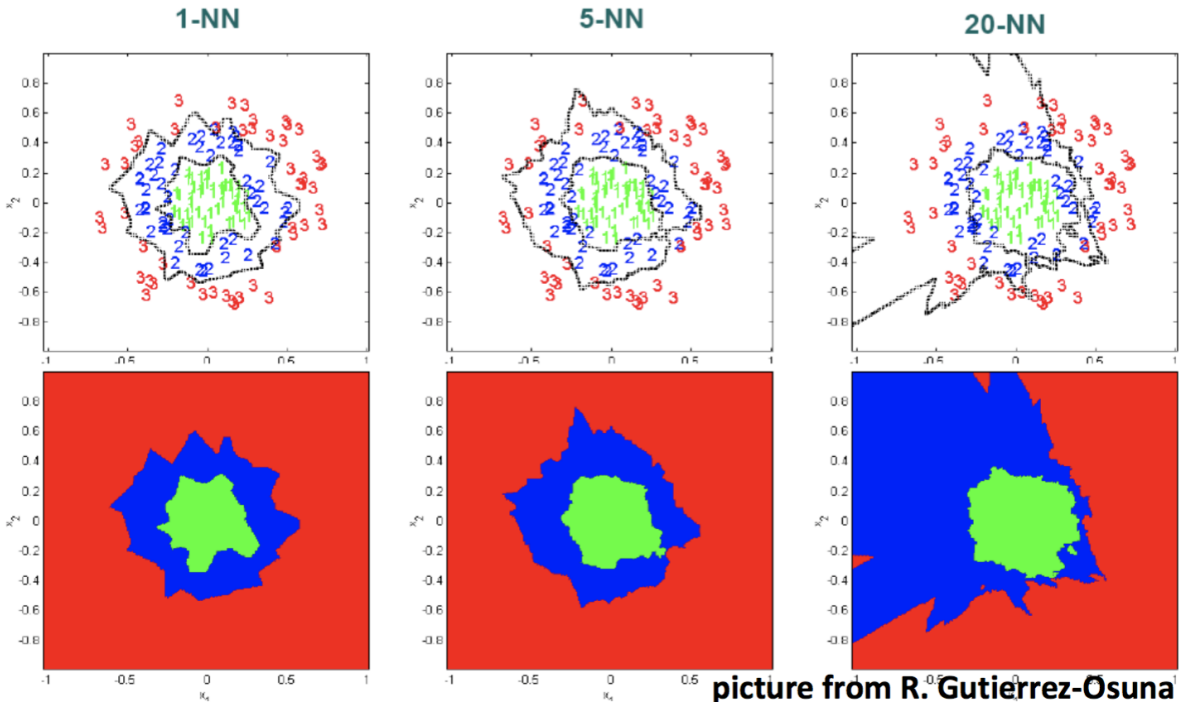
# How to Choose $k$ (Cont.)

- When $k = 1$, the decision boundary is overly flexible and this corresponds to a classifier that has low bias but very high variance.
- $k = 1$ is often used for efficiency, but sensitive to "noise".



# How to Choose $k$ (Cont.)




picture from R. Gutierrez-Osuna

- As $k$ grows, the method becomes less flexible and produces a decision boundary that is close to linear.
- This corresponds to a low variance but high bias classifier.
- Can choose $k$ through cross-validation (coming soon).

# Selection of Distance

- So far, we assumed we use *Euclidian Distance* to find the nearest neighbor:

$$D(a, b) = \sqrt{\sum_k (a_k - b_k)^2}.$$

- Euclidean distance treats each feature as equally important.
- However some features (dimensions) may be much more discriminative than other features.

# Feature Weighting

- Scale each feature by its importance for classification

$$D(a, b) = \sqrt{\sum_k w_k (a_k - b_k)^2}.$$

- Can use our prior knowledge about which features are more important.
- Can learn the weights $w_k$ using cross-validation (coming soon).

# kNN Summary

- **Advantages**
  - Can be applied to the data from any distribution.
  - Very simple and intuitive.
  - Good classification if the number of samples is large enough.
- **Disadvantages**
  - Choosing $k$ may be tricky.
  - Test stage is computationally expensive.
    - No training stage, all the work is done during the test stage.
    - This is actually the opposite of what we want. Usually we can afford training step to take a long time, but we want fast test step.
  - Need large number of samples for accuracy.