Matlab 8: K-Nearest Neighbor Classifiers

Cheng-Hsin Hsu

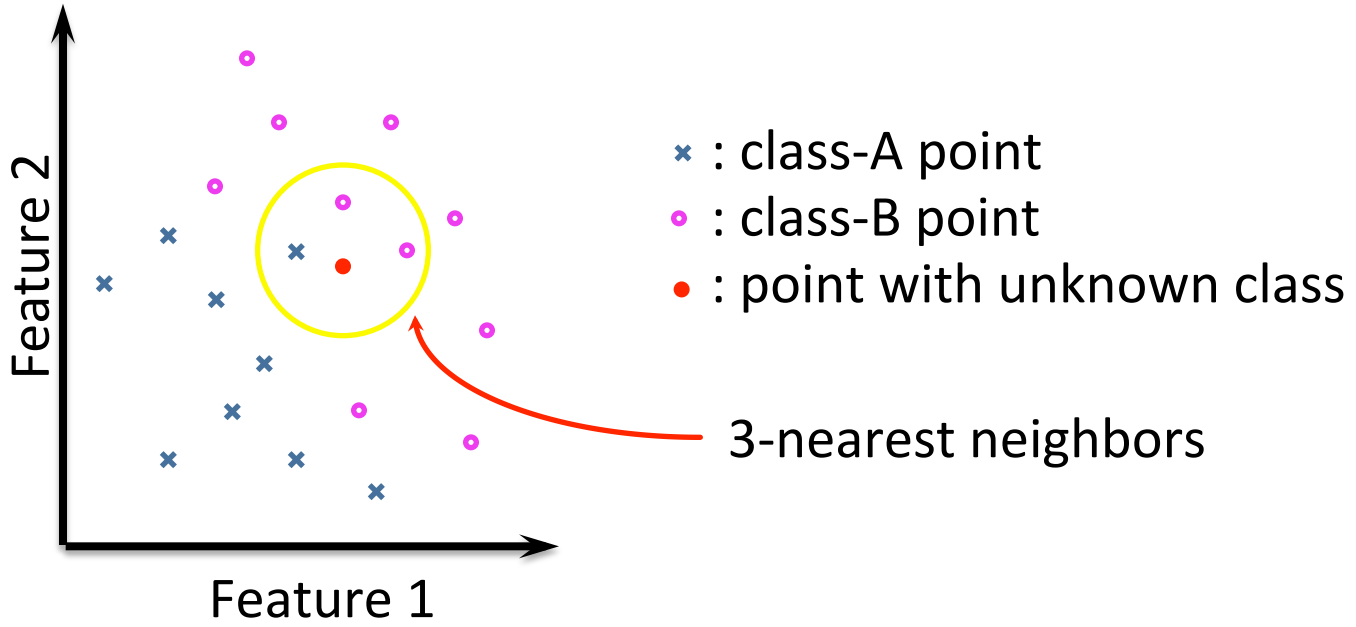
National Tsing Hua University

Department of Computer Science

Slides are based on the materials from Prof. Roger Jang

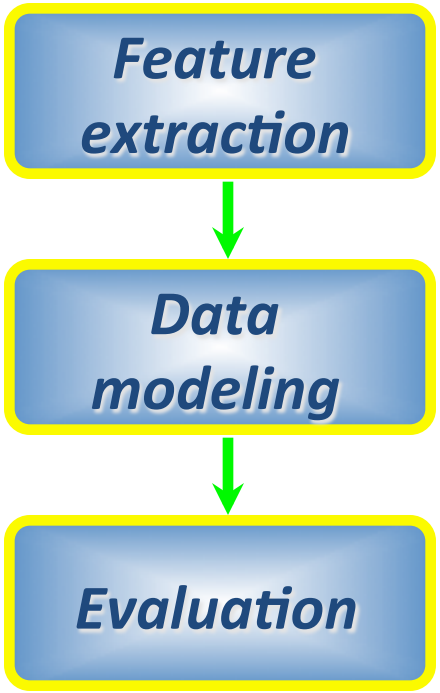
Concept of KNNC

- Steps:
  1. Find the first k nearest neighbors of a given point
  2. Determine the class of the given point by a majority vote among these k neighbors



Flowchart for KNNC

General flowchart of PR:



KNNC:

From raw data to features

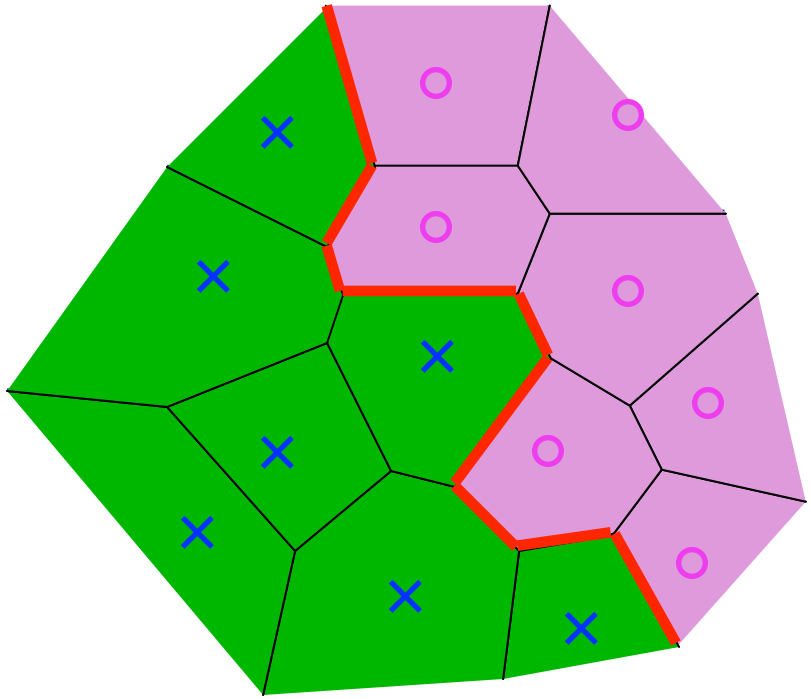
Clustering (optional)

Distance Computation For KNNC

Decision Boundary for 1NNC

Delaunay triangulation: maximize the minimum angle of all the angles of each triangle

Voronoi diagram: piecewise linear boundary



Characteristics of KNNC

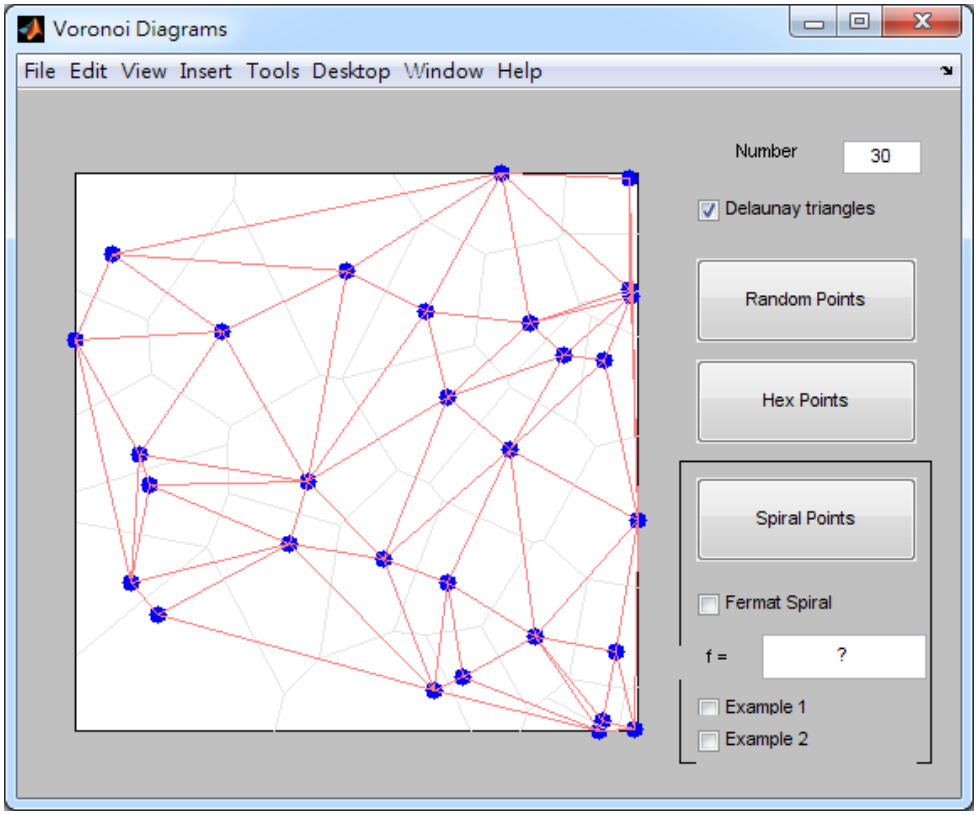
- Strengths of KNNC
  - Very Intuitive
  - No data modeling required
- Drawbacks of KNNC
  - Massive computation required when dataset is big
  - No straightforward way to choose the value of K
  - Rescaling the dataset along each dimension may be tricky

Preprocessing/Variants for KNNC

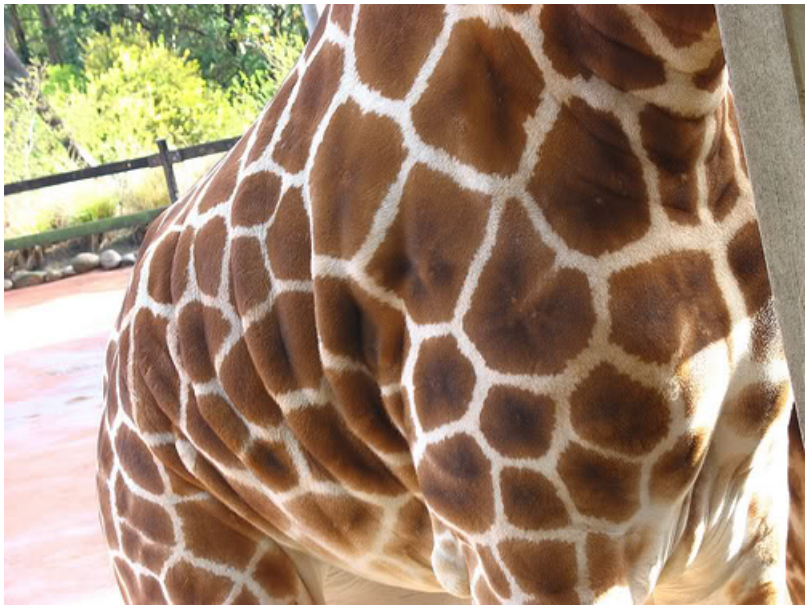
- Preprocessing: $\text{Given } (x_1, x_2, \ldots, x_N), \quad x'_n = \frac{x_n - \mu_x}{\sigma_x} \; \forall n = 1, 2, \ldots, N$
  - Data rescaling to have zero mean and unit variance along each feature
  - Value of K obtained via trials and errors
- Variants:
  - Weighted votes
  - Nearest prototype classification
  - Edited nearest neighbor classification
  - k+k-nearest neighbor

Demos by Cleve

- Delaunay triangles and Voronoi diagram
  - http://mirlab.org/jang/books/dcpr/example.rar



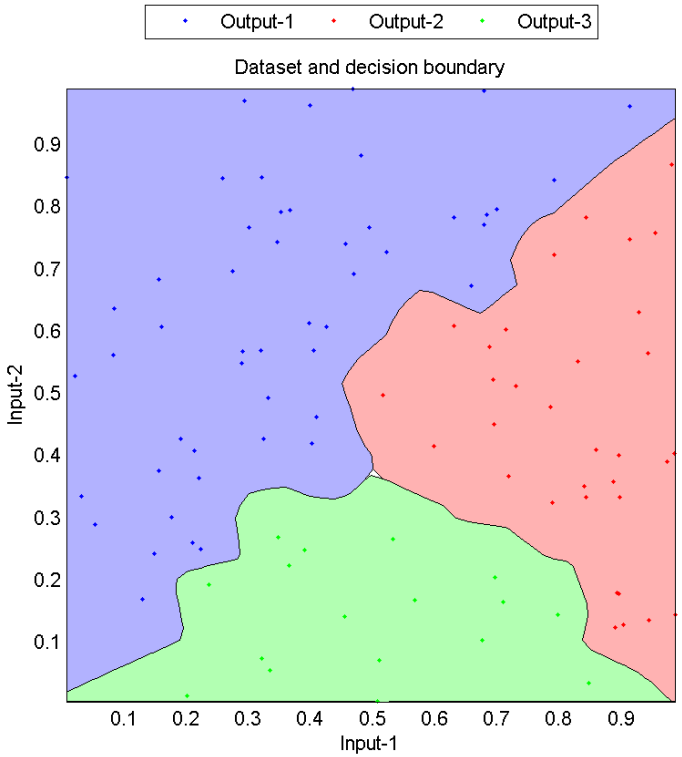
Natural Examples of Voronoi Diagram



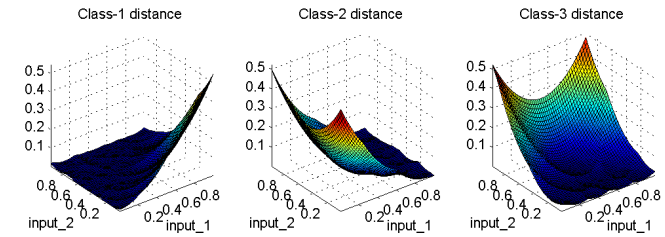
Natural Examples of Voronoi Diagram (cont.)

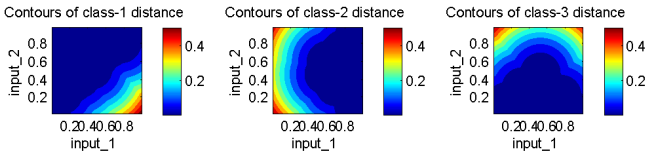
1NNC Decision Boundaries

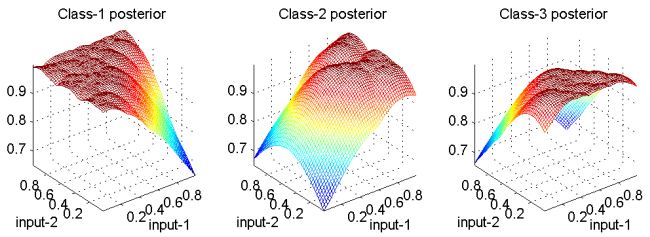
- 1NNC Decision boundaries

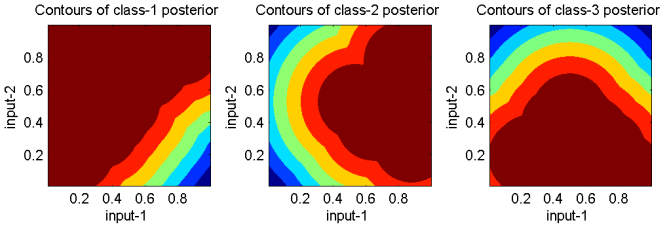


1NNC Distance/Posterior as Surfaces and Contours



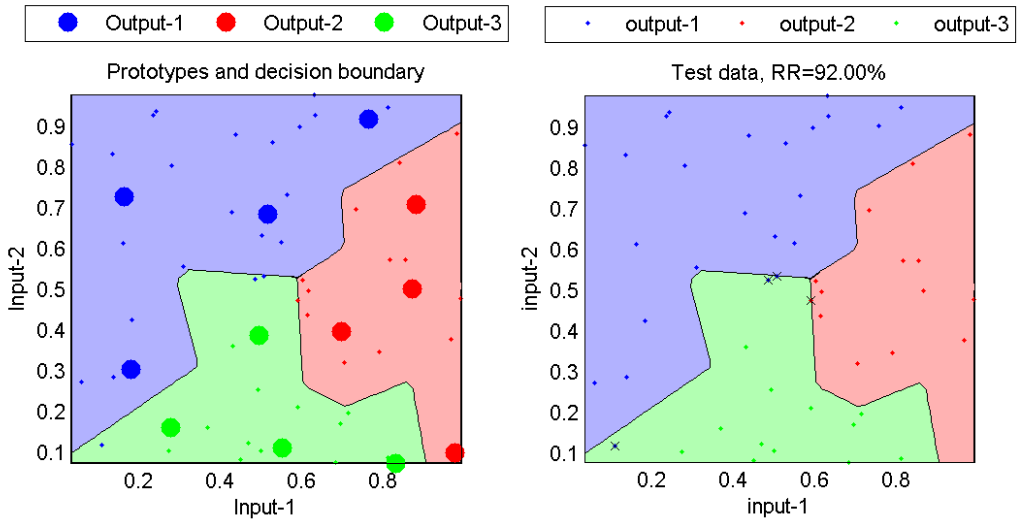






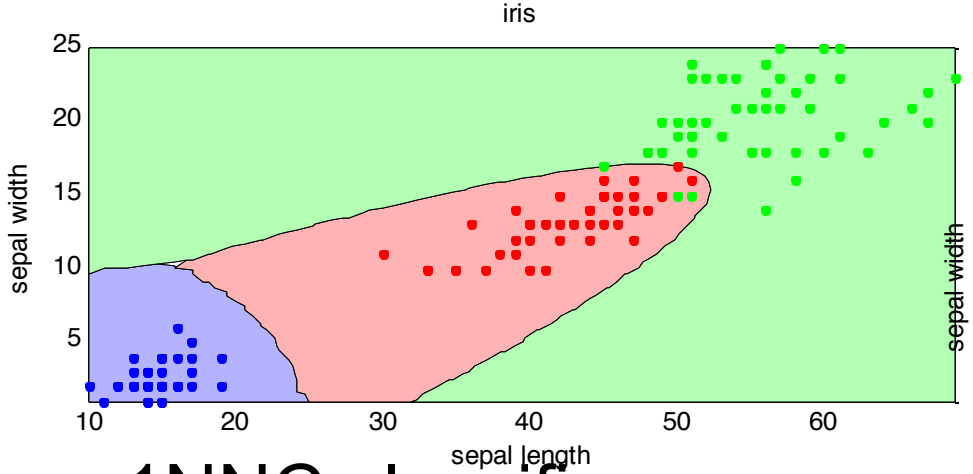
Using Prototypes in KNNC

- No. of prototypes for each class is 4.

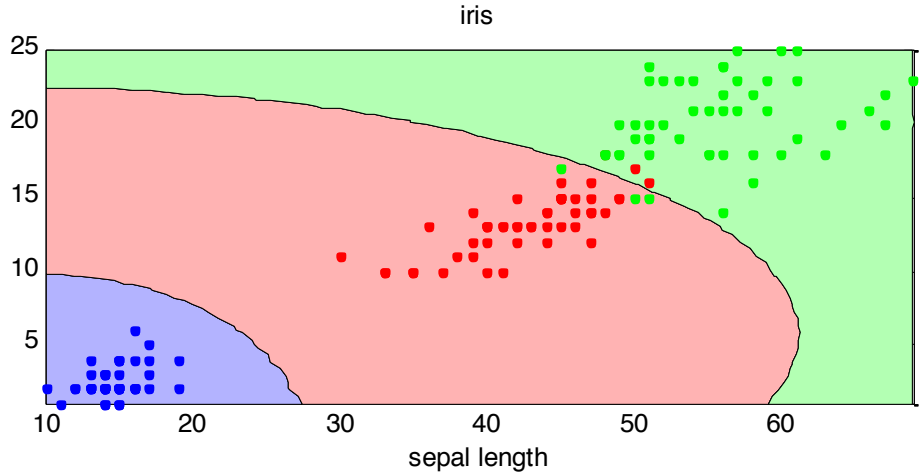


Decision Boundaries of Different Classifiers

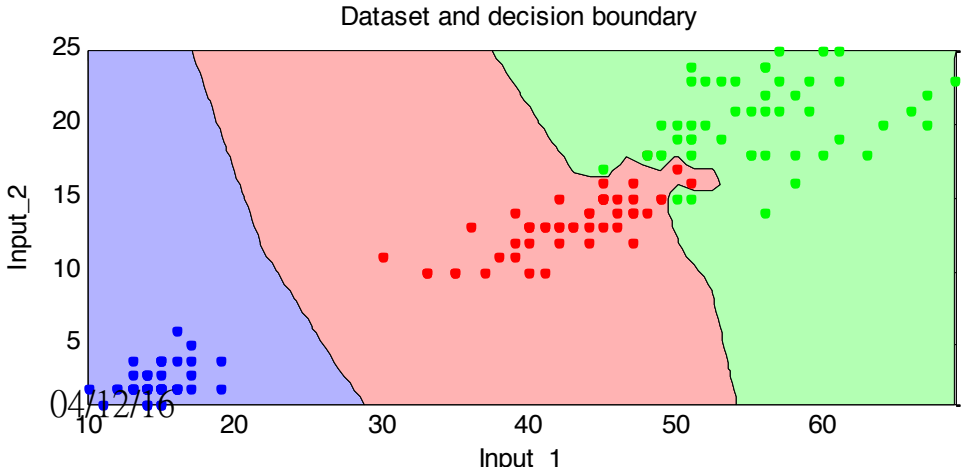
- Quadratic classifier



- Naive Bayes classifier



- 1NNC classifier



Matlab #7 Homework (M7)

1. (1%) Given two sets of 2D points: Red ones at {(0, 1), (2, 3), (4, 5)}; and blue ones at {(2,0), (5, 2), (7,3)}. Please use Euclidean distance in this exercise.
   a) Draw the KNNC decision boundary for K=1. Mark the colors for each point.
   b) Multiple the 6 points' x values by 10. Replot the figure, and explain how the change affects the decision boundary. How would this affect real world problem?
   c) What is the output of 3-NNC of the point (1,2)? If you can add points in the 2D points, in order to flip the classification of (1,2)? Show your work, e.g., where you add the points and the values of them.
   d) What is the complexity of KNNC algorithm? Show your work.

Questions?