k-Nearest Neighbors

CS 540

Yingyu Liang

K-nearest neighbors for classification

- Given training data $\{(x_i, y_i): 1 \leq i \leq n\}$
- Store the training data
- Given a new data point x^*, find its k nearest neighbors in the training data, predict the majority label of the neighbors

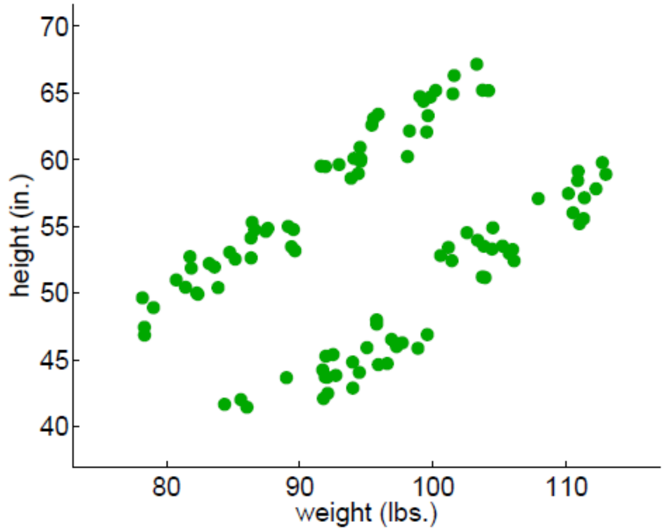
K-nearest neighbors for classification

Input: Training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$; distance function $d()$;
number of neighbors k; test instance $\mathbf{x}^*$

1. Find the k training instances $\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}$ closest to $\mathbf{x}^*$ under distance $d()$.
2. Output y^* as the majority class of $y_{i_1}, \ldots, y_{i_k}$. Break ties randomly.

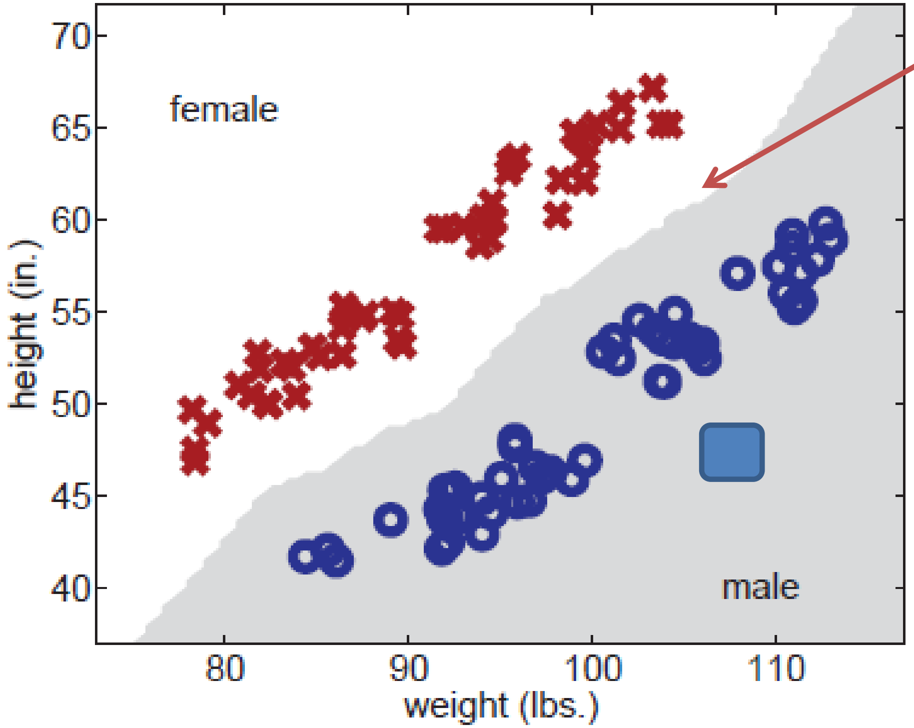
Example: 1-NN for Little Green Man

- Little green men:
  - Predict gender (M,F) from weight, height?
  - Predict age (adult, juvenile) from weight, height?

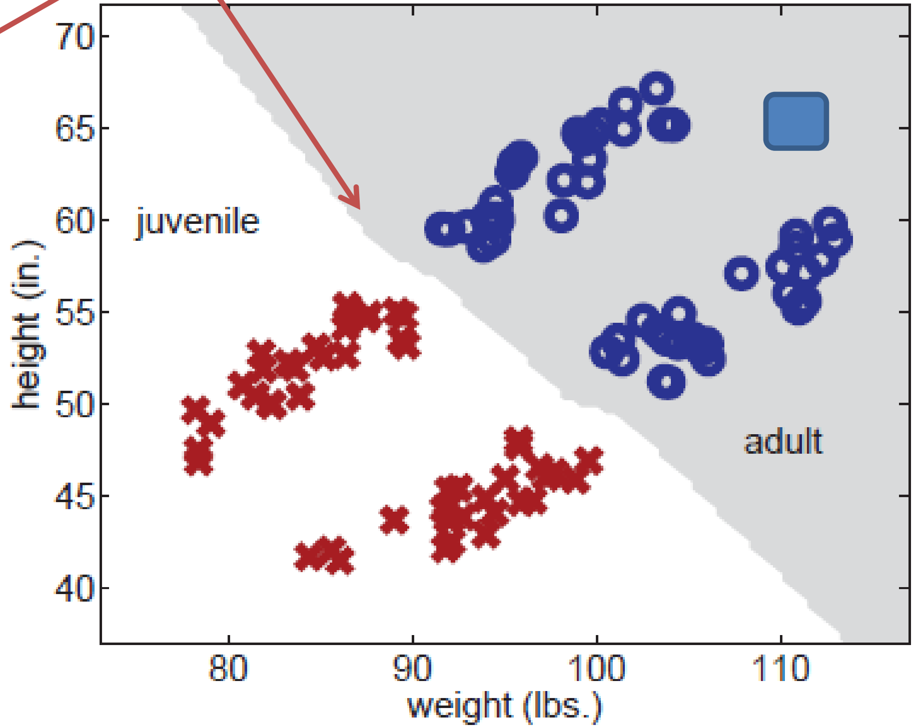


Example: 1-NN for Little Green Man

Decision boundary



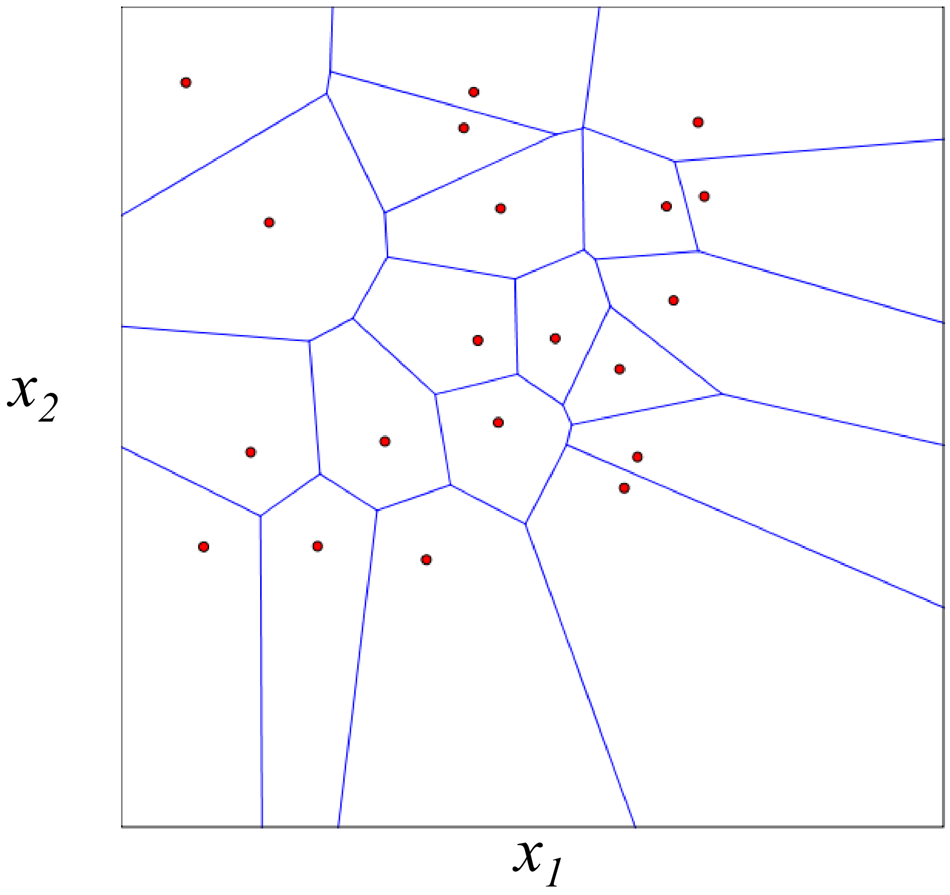
(a) classification by gender



(b) classification by age

The decision regions for 1-NN

Voronoi diagram: each polyhedron indicates the region of feature space that is in the nearest neighborhood of each training instance



K-NN for regression

- What if we want regression?

- Instead of majority vote, take average of neighbors' labels
  - Given test point x^*, find its k nearest neighbors $x_{i_1}, \ldots, x_{i_k}$
  - Output the predicted label $\frac{1}{k}(y_{i_1} + \cdots + y_{i_k})$

How can we determine distance

suppose all features are discrete

- Hamming distance: count the number of features for which two instances differ

suppose all features are continuous

- Euclidean distance: sum of squared differences

  $d(x_i, x_j) = \sum_f (x_{if} - x_{jf})^2$, where x_{if} is the f-th feature

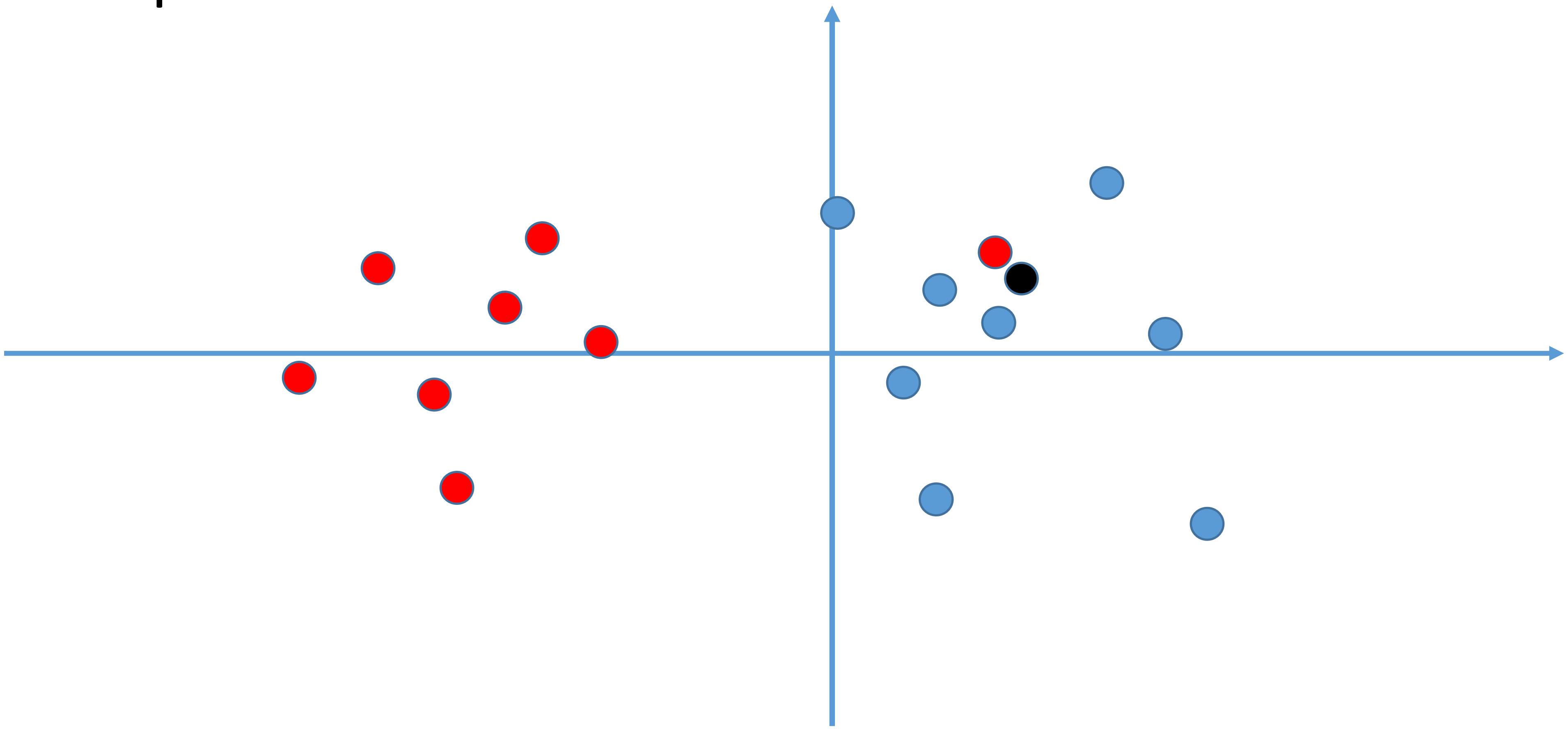
- Manhattan distance:

  $d(x_i, x_j) = \sum_f |x_{if} - x_{jf}|$, where x_{if} is the f-th feature

How to pick the number of neighbors

- Split data into training and tuning sets
- Classify tuning set with different k
- Pick k that produces least tuning-set error

Example: the effect of k



What's the predicted label for the black dot using 1 neighbor? 3 neighbors?