Nearest Neighbor

CSCI 447/547 MACHINE LEARNING

Outline

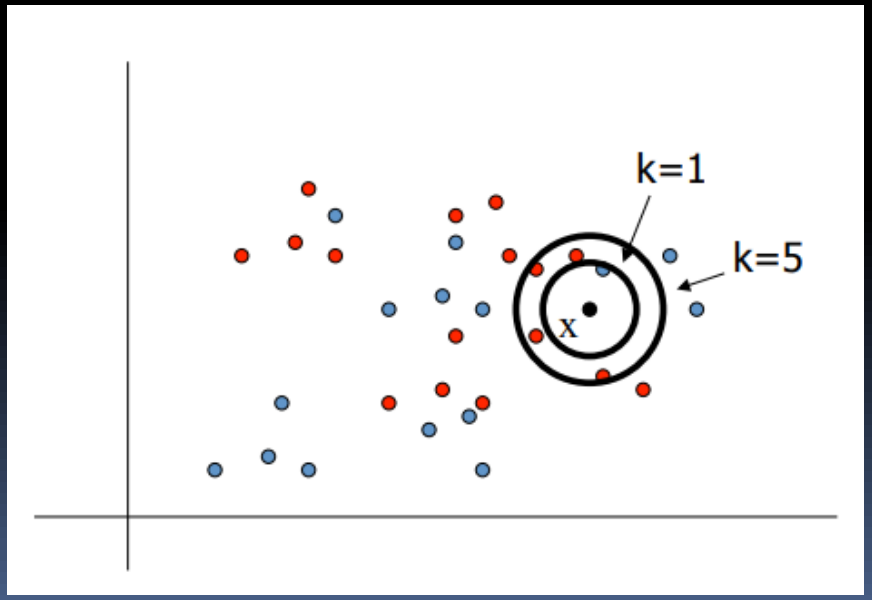
- Nearest Neighbor
  - K-Nearest Neighbor Algorithm
  - Note: Slides were adapted from David Sontag, New York University (who adapted them from Vibhav Gogate, Carlos Questrin, Mehryar Mohri, and Luke Settlemoyer)

Nearest Neighbor

- Supervised learning
- Learning algorithm:
  - Store training examples
- Prediction algorithm:
  - To classify a new example x by finding the training example(x_i, y_i) that is nearest to x
  - Guess the class $y = y_i$

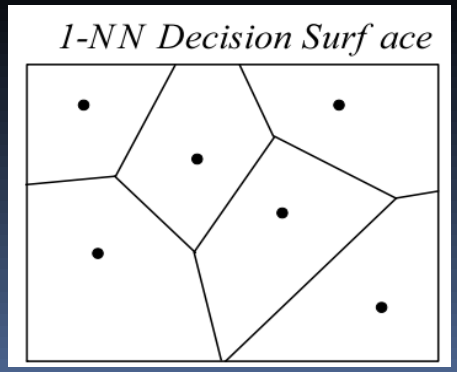
K-Nearest Neighbors Methods

- To classify a new input vector x, examine the k closest training data points to x and assign the object to the most frequently occurring class
  - Common values for k: 3, 5

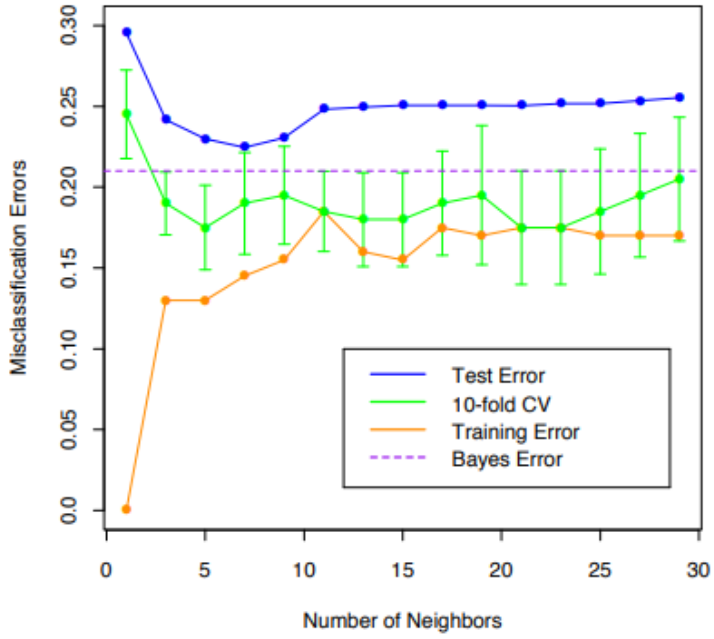


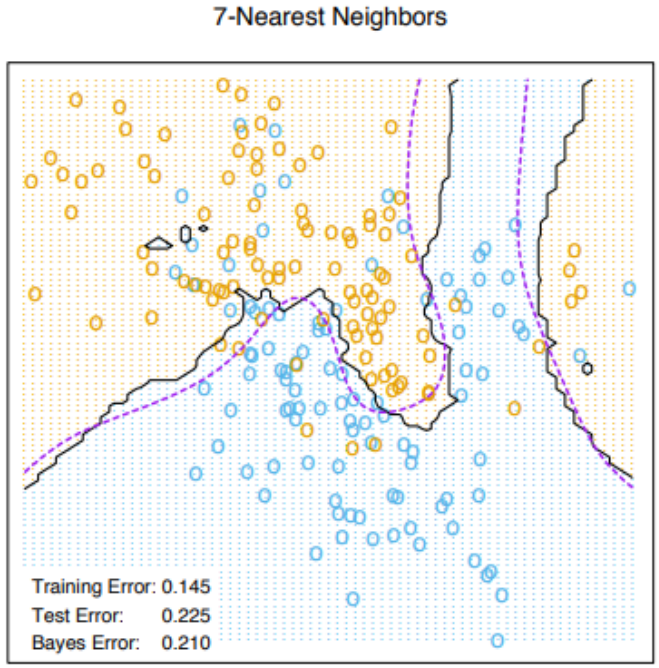
Decision Boundaries

- The nearest neighbor algorithm does not explicitly compute decision boundaries. However, the decision boundaries form a subset of the Voronoi diagram for the training data
  - The more examples that are stored, the more complex the decision boundaries can become



Example Results for k-NN





[Figures from Hastie and Tibshirani, Chapter 13]

Nearest Neighbor

- When to Consider
  - Instance map to points in R^n
  - Less than 20 attributes per instance
  - Lots of training data
- Advantages
  - Training is very fast
  - Learn complex target functions
  - Do not lose information
- Disadvantages
  - Slow at query time
  - Easily fooled by irrelevant attributes

Issues

- Distance measure
  - Most common: Euclidean
- Choosing k
  - Increasing k reduces variance, increases bias
- For high-dimensional space, problem that the nearest neighbor may not be very close at all
- Memory-based technique: Must make a pass through the data for each classification. This can be prohibitive for large data sets.

Distance

- Notation: object with p measurements
- $$x^i = (x_1^i, x_2^i, \ldots, x_p^i)$$
- Most common distance metric is Euclidean distance:
- $$d_E(x^i, x^j) = \sqrt{\sum_{k=1}^{p} (x_k^i - x_k^j)^2}$$
- ED makes sense when different measurements are commensurate – each is variable measured in the same units
- If the measurements are different, say length and weight, it is not clear

Standardization

- When variables are not commensurate, we can standardize them by dividing by the sample standard deviation. This makes them all equally important.
- The estimate for the standard deviation of x_k:
- $$\hat{\sigma}_k = \left(\frac{1}{n}\sum_{i=1}^{n}(x_k^i - \bar{x}_k)^2\right)^{1/2}$$
- Where $\bar{x}_k$ is the sample mean:
- $$\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_k^i$$

Weighted Euclidean Distance

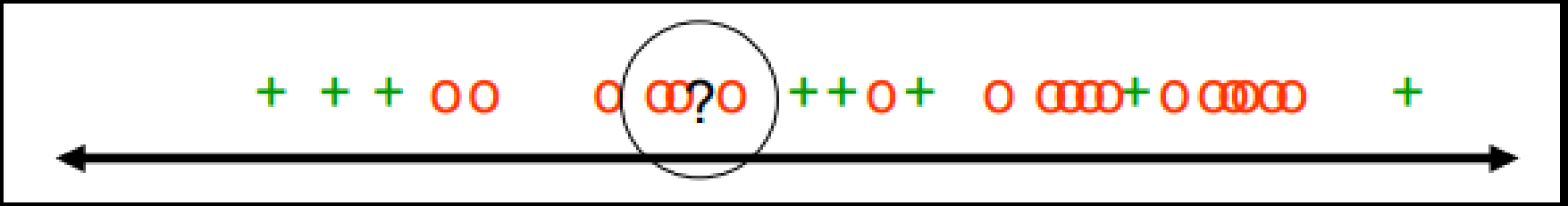
- Finally, if we have some idea of the relative importance of each variable, we can weight them:

$$d_{WE}(x^i, x^j) = \sqrt{\sum_{k=1}^{p} w_k (x_k^i - x_k^j)^2}$$

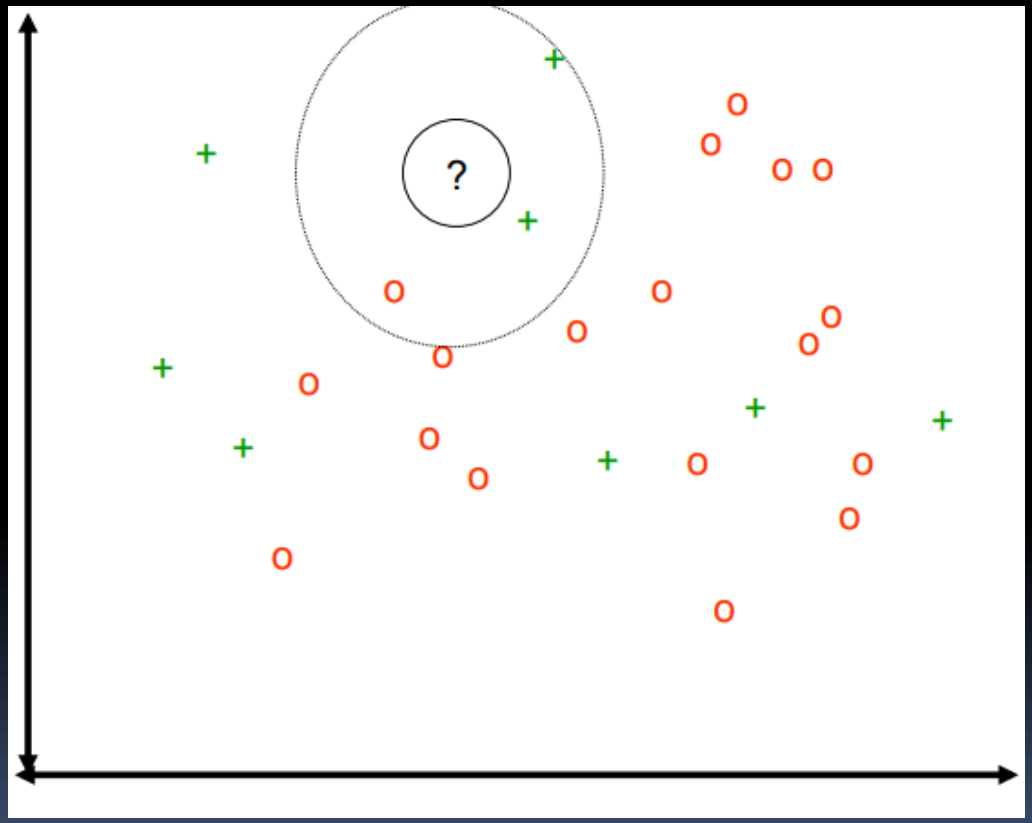
The Curse of Dimensionality

- Nearest neighbor breaks down in high-dimensional spaces because the “neighborhood” becomes very large
- Suppose we have 5000 points uniformly distributed in the unit hypercube and we want to apply the 5-nearest neighbor algorithm
- Suppose our query point is at the origin
  - 1D
    - On a one dimensional line, we must go a distance of 5/5000 = 0.001 on average to capture the 5 nearest neighbors
  - 2D
    - In two dimensions, we must go sqrt(0.001) to get a square that contains 0.001 of the volume
  - ND
    - In N dimensions we must go $(0.001)^{1/N}$

K-NN and Irrelevant Features



K-NN and Irrelevant Features



K-NN Advantages

- Easy to program
- No optimization or training required
- Classification accuracy can be very good; can outperform more complex models

Summary

- Nearest Neighbor
  - K-Nearest Neighbor Algorithm
  - Note: Slides were adapted from David Sontag, New York University (who adapted them from Vibhav Gogate, Carlos Questrin, Mehryar Mohri, and Luke Settlemoyer)