COMP 345: Data Mining
K-Nearest Neighbors

Announcements/Reminders

- **Bring laptop on Wed/Thurs this week**
  - Be prepared to work on your group project.
  - Checkpoint for group project due in 1 week.

- Extra Credit Opportunity (2 points)
  - Attend Dr. Stanley Pounds talk on his Biostatistics Research at St. Jude
  - **Thurs. Nov. 1st at 4pm in Spence Wilson Room**
  - Turn in a 1 paragraph summary of what you learned and what you found interesting about the talk by beginning of class on Mon. Nov. 5th /Tues. Nov. 6th

2

Lazy vs. Eager Learning

- Lazy vs. eager learning
  - **Lazy learning** (e.g., instance-based learning): Simply stores training data (or only minor processing) and waits until it is given a test tuple
  - **Eager learning** (the above discussed methods): Given a set of training tuples, constructs a classification model before receiving new (e.g., test) data to classify
- Lazy: less time in training but more time in predicting
- Accuracy
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form an implicit global approximation to the target function
  - Eager: must commit to a single hypothesis that covers the entire instance space

3

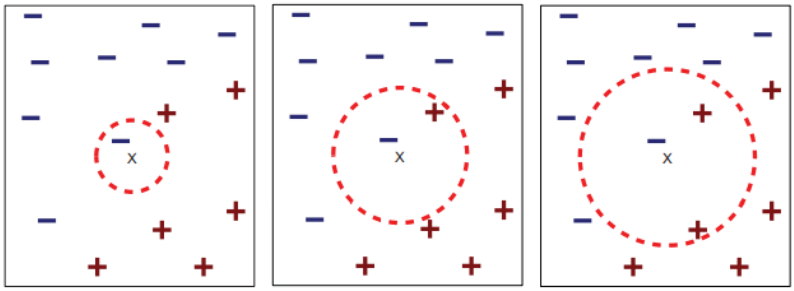
K-Nearest Neighbors

- Amongst the simplest of all machine learning algorithms. No explicit training or model.
- Can be used both for classification and regression.
- Use x's K-Nearest Neighbors to vote on what x's label should be.

4

K-Nearest Neighbors

Classify using the majority vote of the k closest training points.



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

5

K-Nearest Neighbors

K-NN algorithm does not explicitly compute decision boundaries. The boundaries between distinct classes form a subset of the Voronoi diagram of the training data.

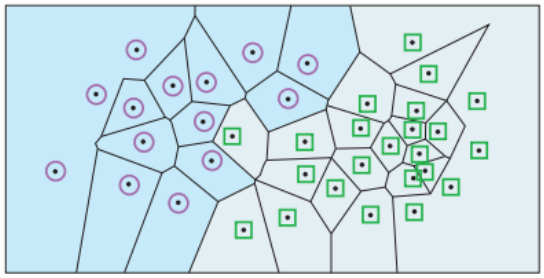


Image by MIT OpenCourseWare.

Each line segment is equidistant to neighboring points.

6

K-NN Algorithm

- Compute the test point's distance from each training point
- Sort the distances in ascending (or descending) order
- Use the sorted distances to select the K nearest neighbors
- Use majority rule (for classification) or averaging (for regression)

7

K-NN: Computing the distances

- The K-NN algorithm requires computing distances of the test example from each of the training examples
- Several ways to compute distances
- The choice depends on the type of the features in the data
- Real-valued features: Euclidean distance is commonly used.
- **Note:** features should be on the same scale.
- **Example:** if one feature has its values in millimeters and another has in centimeters, we would need to normalize

8

K-NN: Some other distance measures

- Binary-valued features
  - Use Hamming distance: $d(x_i, x_j) = \sum_{m=1}^{D} \mathbb{I}(x_{im} \neq x_{jm})$
  - Hamming distance counts the number of features where the two examples disagree
- Mixed feature types (some real-valued and some binary-valued)?
  - Can use mixed distance measures
  - E.g., Euclidean for the real part, Hamming for the binary part
- Can also assign weights to features:

$$d(x_i, x_j) = \sum_{m=1}^{D} w_m d(x_{im}, x_{jm})$$

9

Making k-NN more powerful

- A good value for K can be determined by considering a range of K values.
  - K too small: we'll model the noise
  - K too large: neighbors include too many points from other classes
- There are problems when there is a spread of distances among the KNN.
- Use a distance-based voting scheme, where closer neighbors have more influence.
- The distance measure has to be meaningful – attributes should be scaled
  - Eg. Income varies 10,000-1,000,000 while height varies 1.5-1.8 meters

10

Pros/Cons to K-NN

Pros:

- Simple and powerful. No need for tuning complex parameters to build a model.
- No training involved ("lazy"). New training examples can be added easily.

11

Pros/Cons to K-NN

Cons:

- Expensive and slow: O(nd), n= # examples, d= # dimensions
  - Need to store all training data ***in memory*** even at test time.
- To determine the nearest neighbor of a new point x, must compute the distance to all n training examples. Runtime performance is slow, but can be improved.
  - Pre-sort training examples into fast data structures
  - Compute only an approximate distance
  - Remove redundant data (condensing)

12

Discussion on K-NN Algorithm

- k-NN for real-valued prediction for a given unknown tuple
  - Returns the mean values of the k nearest neighbors
- Distance-weighted nearest neighbor algorithm
  - Weight the contribution of each of the k neighbors according to their distance to the query x_q
    - Give greater weight to closer neighbors

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

- Robust to noisy data by averaging k-nearest neighbors
- Curse of dimensionality: distance between neighbors could be dominated by irrelevant attributes
  - To overcome it, axes stretch or elimination of the least relevant attributes

13

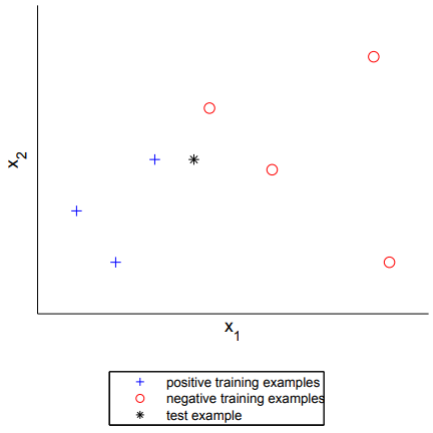
Example

How will * be classified for the following values of k?

K = 1

K = 3

K = 5



14