Data Mining with Weka

Nearest neighbor

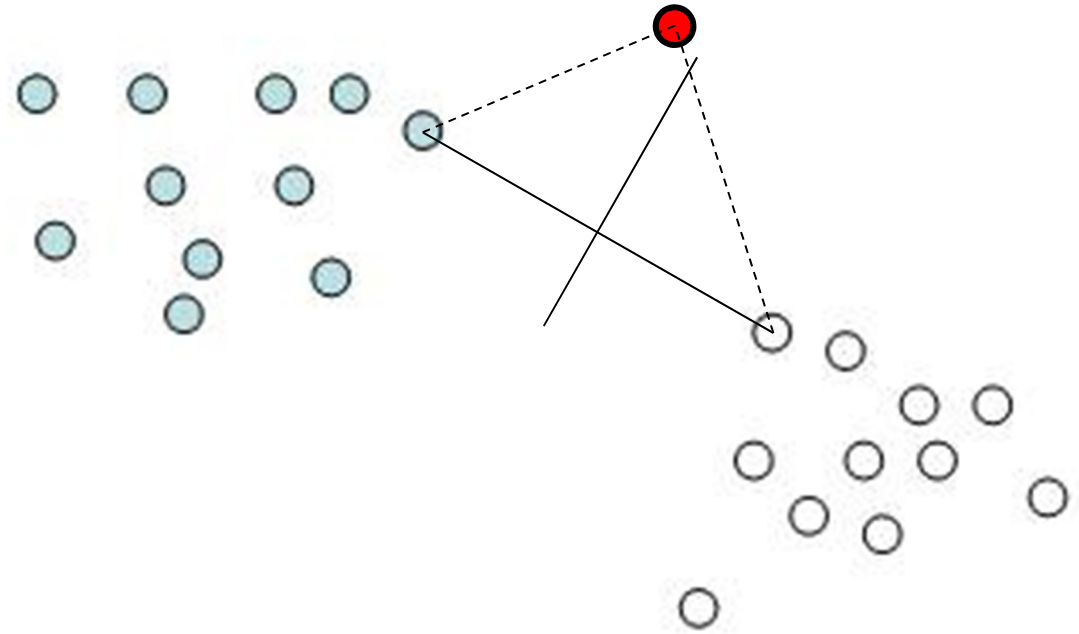
Ian H. Witten

Nearest neighbor

"Rote learning": simplest form of learning

- To classify a new instance, search training set for one that's "most like" it
  - *the instances themselves represent the "knowledge"*
  - *lazy learning: do nothing until you have to make predictions*
- "Instance-based" learning = "nearest-neighbor" learning

Nearest neighbor



Nearest neighbor

Search training set for one that's "most like" it

- Need a similarity function
  - *Regular ("Euclidean") distance? (sum of squares of differences)*
  - *Manhattan ("city-block") distance? (sum of absolute differences)*
  - *Nominal attributes? Distance = 1 if different, 0 if same*
  - *Normalize the attributes to lie between 0 and 1?*

Nearest neighbor

What about noisy instances?

- Nearest-neighbor
- *k*-nearest-neighbors
  - *choose majority class among several neighbors (k of them)*
- In Weka,

  lazy>IBk (instance-based learning)

Nearest neighbor

Investigate effect of changing *k*

- Glass dataset
- lazy > IBk, $k = 1, 5, 20$
- 10-fold cross-validation

$k = 1$	$k = 5$	$k = 20$
70.6%	67.8%	65.4%

Nearest neighbor

- Often very accurate … but slow:
  - scan entire training data to make each prediction?
  - sophisticated data structures can make this faster
- Assumes all attributes equally important
  - Remedy: attribute selection or weights
- Remedies against noisy instances:
  - Majority vote over the *k* nearest neighbors
  - Weight instances according to prediction accuracy
  - Identify reliable "prototypes" for each class
- Statisticians have used *k*-NN since 1950s
  - If training set size $n \rightarrow \infty$ and $k \rightarrow \infty$ and $k/n \rightarrow 0$, error approaches minimum