cs / philo 372

Week 11

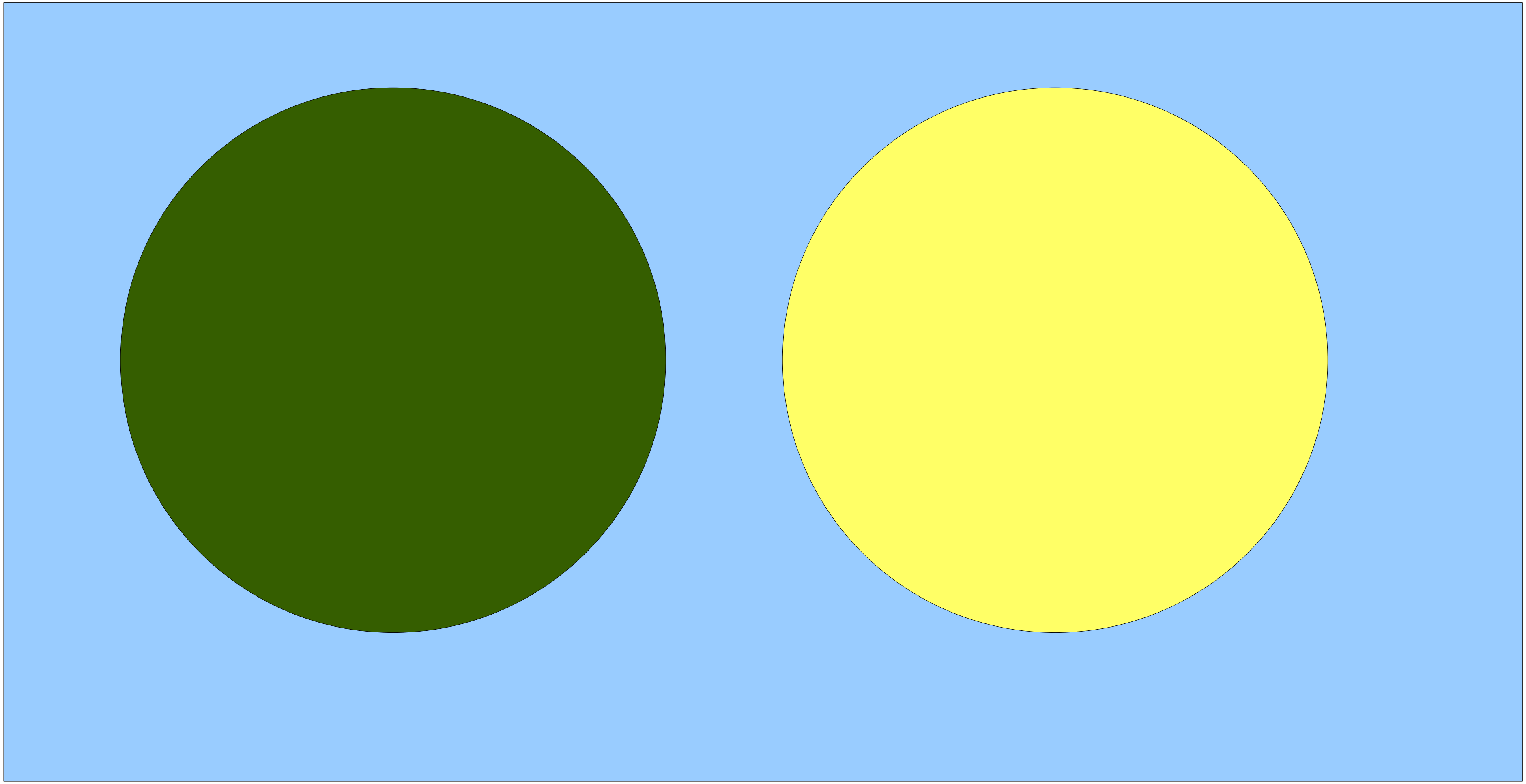
Active Learning
Instance-Based Learning
Neural Network Learning

Active Learning

- Suppose:
  - Algorithm is able to learn one example at a time
  - Examples are free, example labels are expensive
    - any text domain
- Then might make sense to allow learning algorithm to select the examples to label
  - This is "active learning"
  - reported to reduce training set size by factor of 500
- Problem:
  - what examples do you label?
  - Can your learner provide requisite info?

Examples to get labels for

Suppose a 2 category classification problem in 2D.
Further suppose circles represent
locations of examples already seen in each category.



Info Needed from Algorithm

- Indication of "confidence" in label
  - Decision Trees?
  - Decision Lists?
  - AdaBoosted decision stumps?
- Note that binary classifiers can provide a "yes/ no" label and a separate confidence
- Other programs provide a probability statement that can be be interpreted as both a label and a confidence

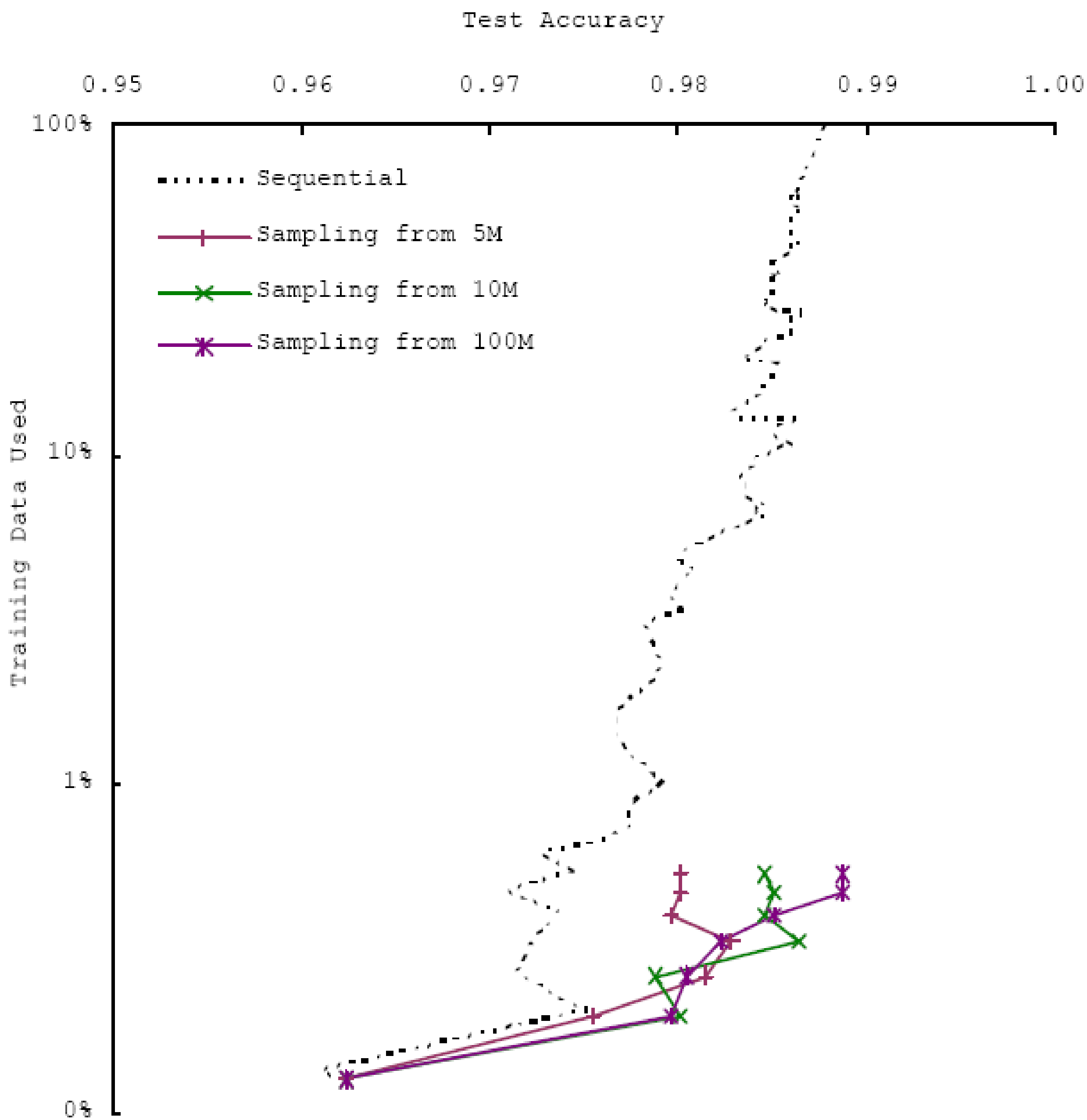
Uncertainty Sampling

Lewis & Catlett (1994)

1. Obtain an initial classifier
2. While expert is willing to label instances
   (a) Apply the current classifier to each unlabeled instance
   (b) Find the b instances for which the classifier is least certain of class membership
   (c) Have the expert label the subsample of b instances
   (d) Train a new classifier on all labeled instances

Uncertainty Sampling

- Problem
  - Use same program to select uncertain as to label.
    - Then program's bias tends to reinforce itself by selecting examples it is uncertain about
    - This can lead to strongly over predicting low frequency classes (among other things)
    - SO??
      - Banko & Brill use committee voting.
        - have committee vote on unlabeled set
          - Committee is formed of learner using different algorithms
        - Take N/2 on which committee agrees least
        - Take N/2 randomly selected (?????)
        - Retrain committee and repeat on smaller unlabeled set



Test Accuracy
0.95
0.96
0.97
0.98
0.99
1.00
100%
10%
1%
0%
Training Data Used
Sequential
Sampling from 5M
Sampling from 10M
Sampling from 100M

Analysis

- About 0.5% of examples is sufficient to achieve accuracy.
- Bigger sets of unlabeled examples improved classification accuracy
  - even though most were never even seen by the classifier
- Committee disagreement does predict errors

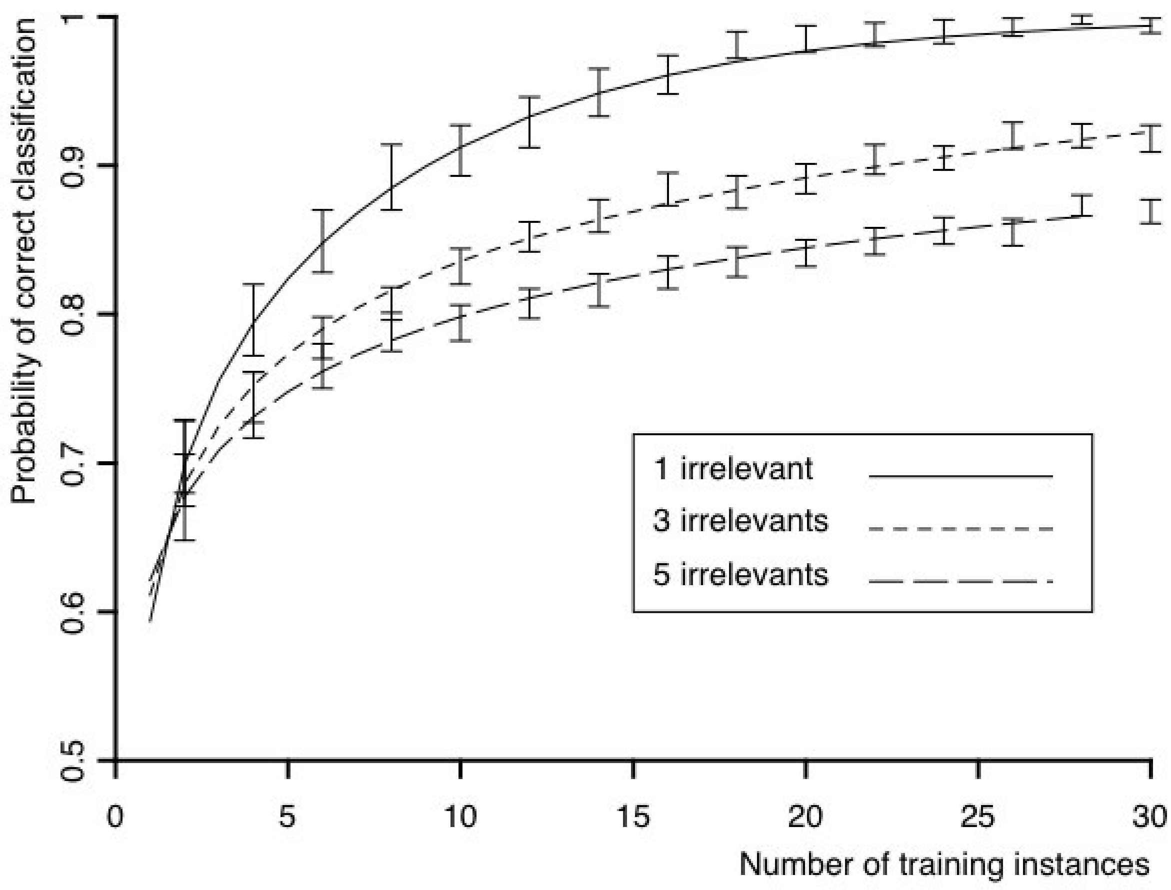
Classifiers In Agreement	Test Accuracy
10	0.8734
9	0.6892
8	0.6286
7	0.6027
6	0.5497
5	0.5000

Instance-Based Learning

- Two general methods
  - Nearest Neighbor
  - Kernel-Based Systems
    - e.g. Radial Basis Functions
- Assumptions
  - Training examples densely sample space
    - at least the interesting parts thereof
  - The classification space is relatively "smooth"

Instance-Based Analysis

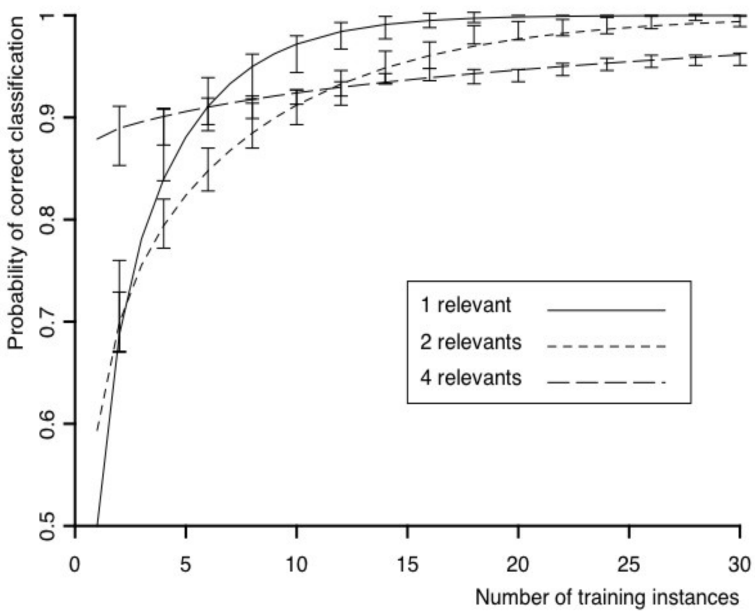
from Langley & Iba 1993



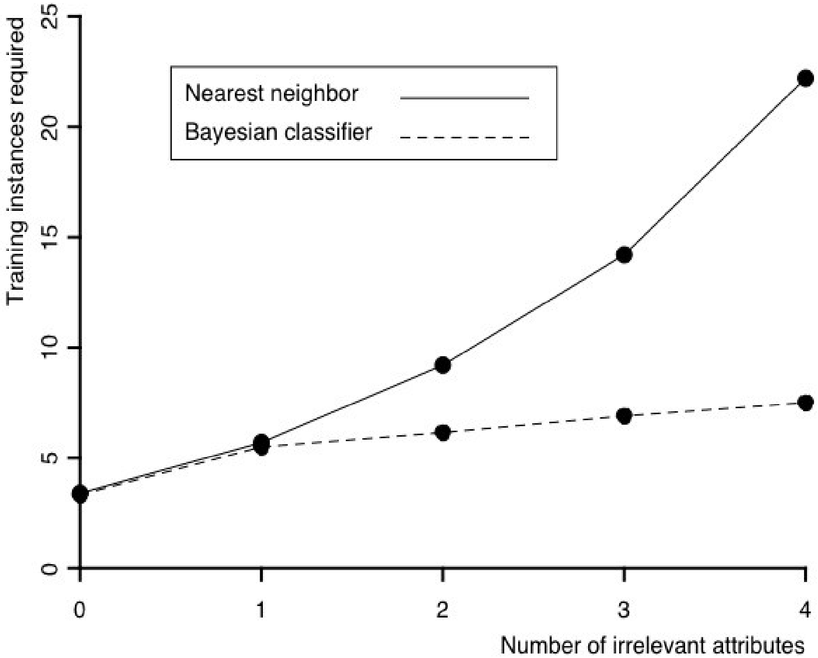
Classification accuracy when there is 1 relevant feature

Instance-Based Analysis

from Langley & Iba 1993



Classification accuracy when there is 1 irrelevant feature

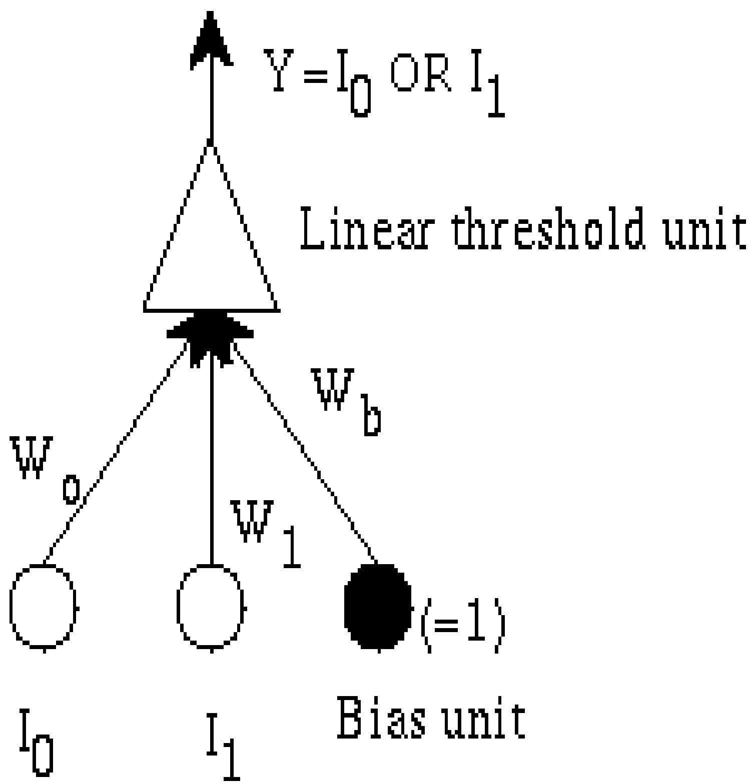


Theoretical accuracy of instance- based methods

Instance-Based Conclusions

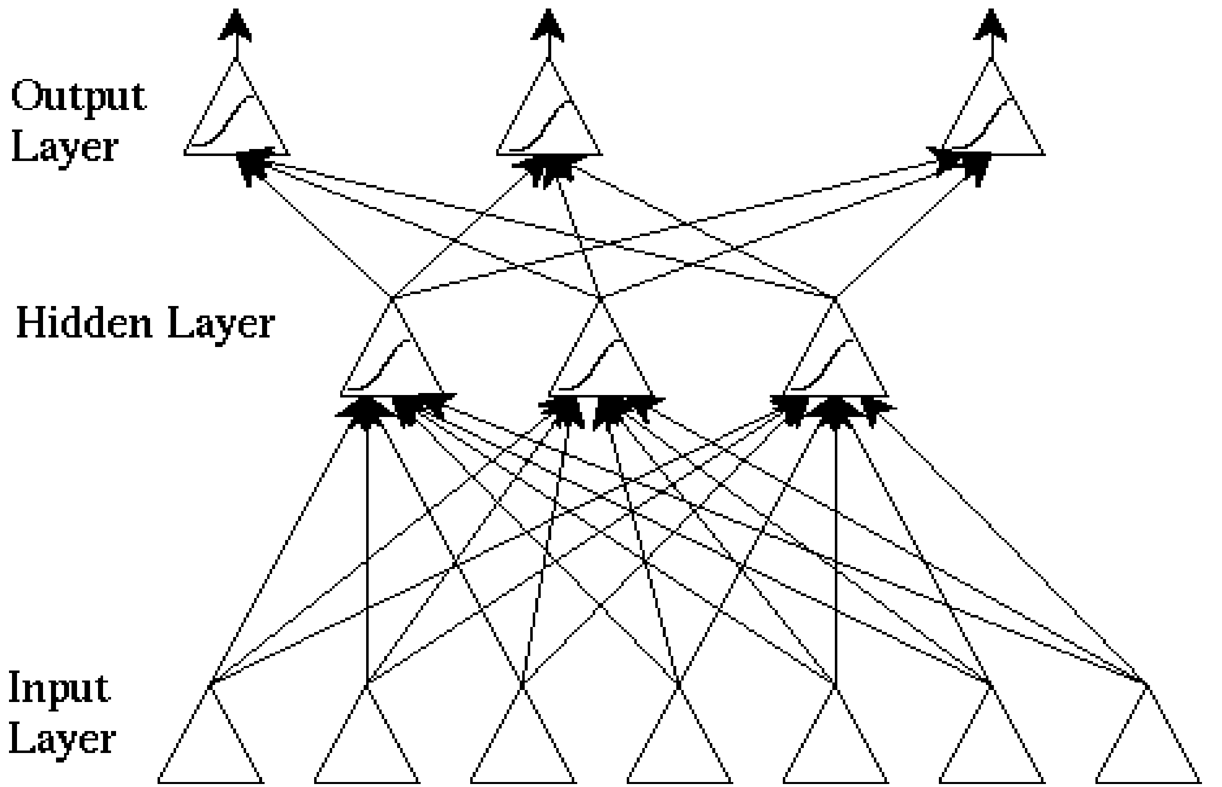
- Apparent simplicity is attractive, but
  - What does "similar" mean
    - L1, L2 ... L-inf norms
    - Hamming distance
    - Mahalanobis distance
  - High dimensional spaces
    - Typically violate "dense sampling" requirement
      - Suppose want to base decisions on K nearest in a hypercube from among N known points in d dimensions
      - Then b^d=K/N where b is size of the length of the cube's side, or b=(K/N)^1/d
      - So if N=1000, d=2, and K=5, then b=0.07
      - But, if N=1,000,000, d=100 and K=5 then b=0.88
  - Irrelevant variables

Neural Networks



- Suppose have 2 inputs (may be binary) and 1 output as at left
- "Linear Threshold Unit"?
- Perceptron learning rule (1963)
  - ch_wi= n*(Y-D)Ii
- Can this network represent all boolean functions?
  - If not, what modifications are needed?
  - What is the "bias unit" for?

Neural Networks



- Rummelhart & McClelland (1983) show that non- linear, differential function can represent and learn all boolean functions
- Xj = 1 / (1+exp(-k*sum(Wi*Xi)))

Neural Networks

- Idea
  - Compute output by computing the “activation” of each node
    - The “forward propagation step”
    - Feedforward, “simple recurrent”, recursive networks
  - With output known compute contribution to error of each input
    - The backward propagation step
    - Can be done through multiple “hidden layers” iff function is differentiable
  - As with perceptron learning rule typically take small steps

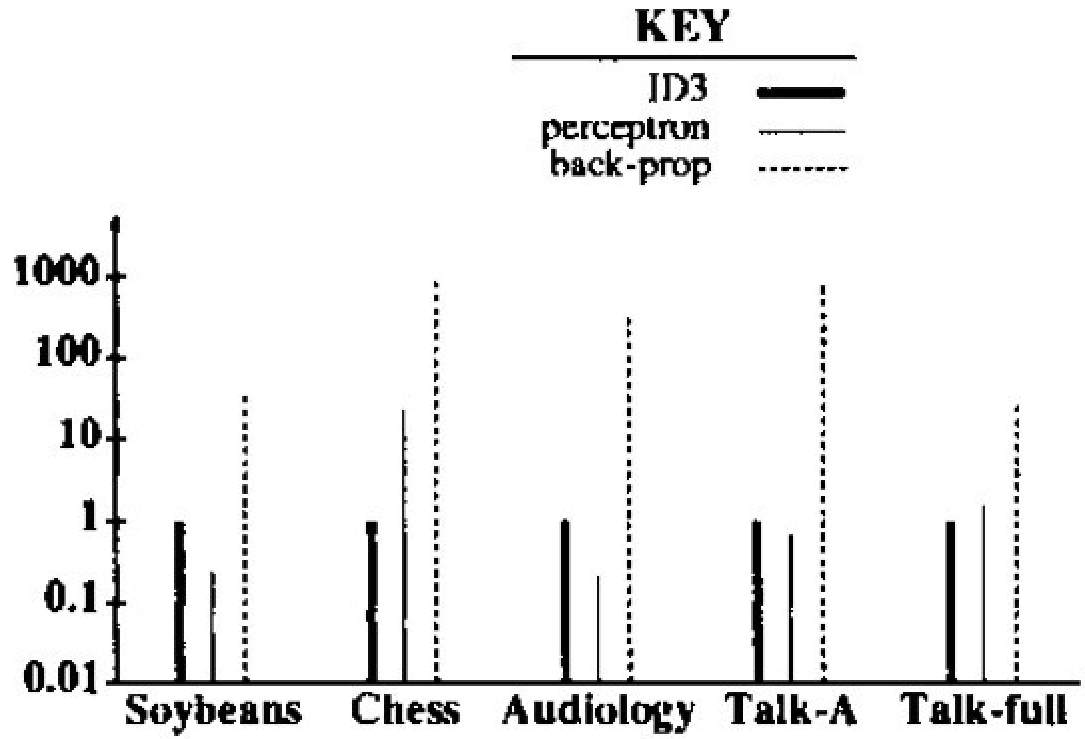


Figure 1. Relative Training Times of the Three Algorithms (scaled to ID3)