Machine Learning, K-Nearest Neighbors and ID Trees: Basic Notes

By Kimberle Koile

(Original date: before Fall 2004)

Basic Definitions

- **Machine Learning** – the process of acquiring a function, based on past inputs and values, that can predict values of future (similar) inputs; use training set to find function, test (aka validation) set to check function's performance.
  - **Supervised** – desired output provided along with input.
    - **Classification** – desired output is a small number of classes.
    - **Regression** – desired output is a continuous variable.
  - **Unsupervised** – desired output *not* provided along with input.
- **Feature** – a descriptor or property used to characterize the input for learning; input is typically a vector of features
- **Feature Space** – space where feature values define the coordinate axes; input vector for each instance defines a point in feature space
- **Cross Validation** – split sample data into N subsets, use each subset as test set, rest as training set; use average and standard deviation of performance on test sets to characterize prediction performance.

K-Nearest Neighbors

Training – Store all feature vectors in the training set, along with each class label.

Prediction – Given a query feature vector, find "nearest" stored feature vector and return the associated class.

"Distance" = $\sqrt{w_1(v_{a1} - v_{b1})^2 + w_2(v_{a2} - v_{b2})^2 + \ldots + w_n(v_{an} - v_{bn})^2}$

v_{a1} is the value of feature 1 in vector a
v_{b1} is the value of feature 1 in vector b
…
w_n is the weight for feature n

"Distance" = $\vec{v}_a \cdot \vec{v}_b$

Normalization? To separate values clustered close together, divide by standard deviation
Relevant features? All features used; to find relevant ones, have to cross validate, dropping features out.
What's the K? Can find best value using cross validation
Voting for vectors? K nearest neighbors vote on class for query feature vector; reduces sensitivity to noise

Identification Trees

Training – Divide feature space into boxes that have uniform labels. Split recursively along each axis to define a tree.

Average disorder = $\sum_b \left(\frac{n_b}{n_t}\right) \times \left(\sum_c -\frac{n_{bc}}{n_b} \log_2\left(\frac{n_{bc}}{n_b}\right)\right)$

n_b is the total number of samples in branch b
n_t is the total number of samples in all branches
n_{bc} is the total of samples in branch b of class c

Prediction – Test features of query feature vector according to identification tree generated during training, return class at leaf of tree.

Relevant features? Irrelevant features are ignored because have large disorders.
Whose Razor? Occam's: The world is inherently simple. Choose smallest consistent tree.
Why greedy? Finding simplest tree is intractable; greedy search using minimum average disorder as heuristic.

Entropy: $E = -a \log_2 a - b \log_2 b - c \log_2 c \ldots$