Homework 2

October 11, 2018

- Chapter 3
  1) Give it some thought: #3, #4
     Remark: In problem 4, we define irrelevant attributes as follows. Suppose that the domain has m attributes. Denote
     $$\mathcal{S} = \{1, 2, \ldots, m\}.$$
     We represent an example by an m-dimensional vector of the form
     $$\boldsymbol{x} = (x_1, x_2, \ldots, x_m).$$
     Let s be a subset of $\mathcal{S}$. We define the distance between attribute vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ using the attributes in set s. Particularly, define
     $$d_s(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in s} d_i(x_i, y_i), \tag{1}$$
     where $d_i(x_i, y_i)$ is the distance for the i-th attribute between x_i and y_i. Let $h(\boldsymbol{x}, s)$ be the class of example $\boldsymbol{x}$ determined by a 1-NN classifier using distance measure d_s in (1). Let $c(\boldsymbol{x})$ be the class of example $\boldsymbol{x}$. We say that attributes in subset s are relevant, if the number of examples in the training set that are correctly classified using distance measure $d_{\mathcal{S}}$ is **less than or equal to** that correctly classified using distance measure d_s. We say that attributes in $\mathcal{S} - s$ are irrelevant attributes.
  2) Computer assignment: #1, #2
     Remark: Use the data set in iLMS.

Due date: Thursday, Oct. 18, 2018 (Note: You can submit the homework in class on the due date. Alternatively, you can submit your homework to Room 845 EECS building before 5 pm on the due date. No late homework is accepted.)

Give It Some Thought

3. Design an algorithm that uses hill-climbing search to remove *redundant examples*. Hint: the initial state will contain the entire training set, the search operator will remove a single training example at a time (this removal must not affect behavior).
4. Describe an algorithm that uses hill-climbing search to remove *irrelevant attributes*. Hint: withhold some training examples on which you will test 1-NN's classifier's performance for different subsets of attributes.

Computer Assignments

1. Write a program whose input is the training set, a user-specified value of k, and an object, $\mathbf{x}$. The output is the class label of $\mathbf{x}$.
2. Apply the program implemented in the previous assignment to some of the benchmark domains from the UCI repository.[5] Always take 40 % of the examples out and reclassify them with the 1-NN classifier that uses the remaining 60 %.