CS478 Machine Learning
Spring 2000
Assignment 4

Due electronically by Tuesday, April 11, 11 a.m.

Problem Set Portion (50 pts)

1. Exercise 4.7 in *Mitchell* (10 pts).
2. Exercise 4.9 in *Mitchell* (10 pts).
3. Genetic Algorithms (15 pts)

   Consider the task of assigning T tasks on m processors so that the time until the last task finishes is minimized. Assume that each task t takes $l(t)$ time to run. In addition, assume there is a partial order on the tasks, such that if $t_1 < t_2$ (i.e., task t_1 precedes task t_2 in the partial ordering) then task t_1 must complete before t_2 can begin. You can assume that all runtimes are integers.

   Specify a genetic algorithm solution to this problem:

   (a) Describe what the *individuals* in the population will represent.
   (b) Define a suitable *fitness function* for this problem.
   (c) Specify an algorithm for the *mutation* operation.
   (d) Specify an algorithm for the *crossover* operation.
4. Exercise 9.4 in *Mitchell* (15 pts).

Programming Portion (30 "correctness" pts + 20 "style and documentation" pts)

1. Implement the instance-based learning (k-nearest neighbor) algorithm described in class and in Section 8.2 of the *Mitchell* text and apply it to the data sets used for assignment 1 (TICTACTOE and MUSHROOM2). You may use C, C++, Java, Lisp, or Scheme to implement ID3. If you would like to use any other language, contact Prof. Cardie for approval first.[1]

   - As in homework 1, your program should **prompt the user for a training file and a test file** or allow both to be specified in the command line invocation of your program.
   - Your code should **prompt the user for the value of** k or allow k to be specified in the command line invocation of your program.
   - Implement **Hamming distance** as the distance measure. For the Hamming distance, the distance between two features is 0 if their values are identical and 1 if they are different.
   - If there are more than k training instances with the same distance to the test instance **allow all of the "ties" to participate in the the classification decision**. If $k = 1$, for example, and the closest case to the test instance is at distance d but there are three such nearest neighbors, let all three cases participate in the classification decison.

[1] For C and C++ prorammers, we encourage the use of *gcc*, the GNU compiler. For Java programmers, we encourage the use of SUN's JDK 1.2.

- **Use majority vote** among the retrieved cases to determine the class of the test instance. Make an arbitrary or random choice if there are still ties after the majority vote.
- No special indexing of the cases is required for this assignment.

You will need to turn in a **Trace File** showing the performance of your instance-based learning algorithm on two training and test sets.

Generating a Trace File

(a) Print the names of the training and test sets.

(b) Print the value of k.

(c) Print the training and test set sizes.

(d) For the first 10 test instances, print

i. the name of the test instance

ii. the number of nearest neighbors (remember that in cases of ties, there can be more than k nearest neighbors)

iii. the names of and distances to these k-nearest neighbors (plus any ties)

iv. the predicted class for the test instance as determined by the instance-based learning algorithm

v. the correct class of the test instance (as indicated in the test set)

(e) Print the accuracy of the k-nn algorithm on the test set.

Each of these should be clearly labeled in the output.

Electronic Submission

All of the files for this assignment should be submitted electronically as a single **zip archive** or **tar** file. Electronic submission is available at the CS478 home page.

Files to Turn In

1. The **source code** and **executable(s)** for the instance-based learning program. Include all files that go with it, such as .h files. If you are using C or C++, please include a **Makefile**. The extension of the source files should indicate the programming language in which the code was written (e.g. ibl.c or ibl.scheme).

2. A **README** file that includes information on how to run your program.

3. A **trace of your instance-based learning program** on the two data sets listed below using both $k = 1$ and $k = 3$. The data sets can be downloaded from the CS478 web page (see Assignment 1).

   (a) tictactoe.train
   tictactoe.test

   (b) mushroom2.train
   mushroom2.test

   Trace files should be named with a ".trace" extension.

4. Your answers to the problem set. Our preference is that you turn in the answers to the problem set on paper. However, the answers to the questions may be turned in electronically as an ASCII text file (**assign4.txt**), a postscript file (**assign4.ps**), or a pdf file (**assign4.pdf**). Other file formats will not be accepted.