03-60-574
Machine Learning
Take-Home Examination
Deadline: December 15^{th} by noon

Dr. Alioune Ngom

December 3, 2006

Question #1 (20 points) Consider applying Genetic Algorithms to the task of finding an appropriate set of weights for an artificial neural network (in particular, a feed-forward network identical to those trained by *back-propagation*). Consider a $3 \times 2 \times 1$ layered, feed-forward network. Describe an encoding of network weights as a bit string, and describe: an appropriate set of cross-over operators, and, possible fitness functions. State one advantage and one disadvantage of using Genetic Algorithms in contrast to back-propagation to train network weights.

Question #2 (10 points) Consider two perceptrons defined by the threshold expression $w_0 + w_1x_1 + w_2x_2 > 0$. Perceptron A has weights values

$$w_0 = 1, w_1 = 2, w_2 = 1$$

and perceptron B has weights

$$w_0 = 0, w_1 = 2, w_2 = 1$$

Is A *more-general-than* B?

Question #4 (10 points) Consider again the example application of Bayes rule given in lecture #6 (at the beginning). Suppose the doctor decides to order a second laboratory test for the same patient, and suppose the second test returns a positive result as well. What are the posterior probabilities of *cancer* and $\neg$*cancer* following these two tests? Assume that the two tests are independent. What is in your opinion, the difference between Bayesian Learning and Decision Tree Learning?

Question #5 (10 points) Draw the Bayesian belief network that represents the conditional independence assumptions of the Naive Bayes Classifier for the *PlayTennis* problem of lecture #6 (page 19). Give the conditional probability table associated with the node *Wind*.

Question #6 (10 points) Suppose you have two learning algorithms, and you want to compare their accuracy. Discuss, in a point format, an empirical protocol to carry out such a comparison.

Question #7 (20 points) The *curse of dimensionality* is a major limitation (i.e. problem) in using *instance-based learning algorithm* (such as *k-nearest neighbor*) for learning tasks that contain a large number of attributes (i.e. dimension). Methods to overcome the *curse of dimensionality problem* are based on determining *all* (or, a subset of) the attributes that are relevant for the classification of instances. This reduces the number of attributes to a comfortable level for the k-NN algorithm to function well. Invent a Genetic Algorithm approach for determining the *best* set of relevant attributes needed for use in a k-NN learning algorithm. Assuming we have $n = 100$ attributes, clearly describe the problem representation and the fitness function, appropriate for this problem.

Question #8 (10 points)

1. Why does information gain prefer attributes with many different values? What can that lead to? What is the remedy?
2. We normally do not test the same attribute twice along the same path in a Decision Tree. Why?