Assignment 6

CSU520, Spring 2008
Due: Wednesday, April 16
Note: This assignment will not be accepted late.

Part I. Pencil-and-paper exercises.

1. Hand-design decision trees to represent the following boolean functions:

- $A \wedge \neg B$
- $A \vee (B \wedge C)$
- $(A \wedge B) \vee (C \wedge D)$

2. (a) Hand-design a 2-input perceptron that implements the boolean function $A \vee \neg B$. (b) Hand-design a 2-layer (i.e., a 1-hidden-layer) network of perceptrons that implements $A \text{ XOR } B$.

3. Use a naive Bayes classifier with the PlayTennis data (which can be found in `PlayTennisData.pdf` in the `programs/learn` subdirectory of the course web site) to classify the attribute vector `(Outlook = sunny, Temperature = mild, Humidity = high, Wind = strong)`. That is, determine the conditional probabilities of `PlayTennis = yes` and `PlayTennis = no` given this instance under the naive Bayes conditional independence assumptions and then determine which value the corresponding classifier gives to `PlayTennis`.

Part II. Computer exercises

For this part you will not need to write any programs of your own. You should use the programs and other files found in the `programs/learn/learn-progs` subdirectory of the course web site. (This entire subdirectory is available as the gzipped tar file `learn-progs.tar.gz`.) Consult the file `README` in that subdirectory to get an idea of what's there and how to run it. (If you have any additional questions about these programs, feel free to ask me as well.)

4. Run the decision tree learning program on the PlayTennis data of Problem 3. (This data, suitably formatted for all the learning programs, is found in the file `play-tennis-data.lisp`.) Draw the tree generated by the program and under each leaf list all the training instances that end up at that leaf. To see the tree generated by the program, evaluate `(show-decision-tree)` after running the decision tree on this data. You will have to figure out for yourself which training instances correspond to which leaf.

Finally, run the program to determine what classification is given to `PlayTennis` for the instance `(Outlook = sunny, Temperature = mild, Humidity = high, Wind = strong)`. by this decision tree.

5. Run the perceptron algorithm on the same PlayTennis data (by using the same file `play-tennis-data.lisp`). The program automatically uses a 1-out-of-3 encoding for each of the 3-valued attributes Outlook and Temperature, while using a single node for each of Humidity and Wind. Is the data linearly separable using this representation?

Then run the program to determine what classification is given to `PlayTennis` for the instance `(Outlook = sunny, Temperature = mild, Humidity = high, Wind = strong)`. by this perceptron.

6. Run the perceptron algorithm on a different representation of the PlayTennis data, this time using a single node for each attribute, with the Outlook attribute values encoded as Sunny = 0, Overcast = 1/2, and Rain = 1, while Temperature is encoded as Cool = 0, Mild = 1/2, and Hot = 1. (To do this, simply use the file `play-tennis-alt-data.lisp`.) Is the data in this form linearly separable?

7. Run the backpropagation algorithm on the same encoding of the PlayTennis data used in the previous problem (using 2 hidden units). Briefly comment on the results and how they compare with the results you got in Problem 6.

In addition, run the program to determine what classification is given to `PlayTennis` for the instance `(Outlook = sunny, Temperature = mild, Humidity = high, Wind = strong)`. by this backprop net. (You will have to use the appropriate numerical encodings of `Outlook` and `Temperature` when running this neural network on this attribute vector.)

What you should turn in for Part II:

- hardcopy of all relevant dribble files; and
- appropriate written commentary recording your observations and answers to all the questions above.