Learning and Adaptivity — ST 2014

Prof. Gerhard. Kraetzschmar — Exercise Sheet 2

Distributed on Apr-08 — Due Date: Apr-25

For the following exercises on decision tree and neural network learning, there are two sets of data available: `zoo` and `letter-recognition`. Before using them in the exercises, the participants need to prepare the data sets such that they can be used by the respective tools (see below). The decision tree exercises can be done — at least to a good deal — with the `dTree` applet demonstrated in class, which is available in moodle. We are working on providing alternative software, like `weka` or `scikit-learn` plus tutorial material; stay tuned.

8 **Exercise 7: Decision Tree Learning: Simple Prediction**

Using the `zoo` database learn a decision tree for ..

2 1. ... predicting attribute 18 (type) from attributes 2 to 17,

2 2. ... predicting attribute 18 (type) from attributes 1 to 17,

2 3. ... predicting attribute 6 (airborne) from attributes 2 to 5 and 7 to 18, and

2 4. ... predicting attribute 1 (animal) from attributes 2 to 18.

For each case, use *all* available data, experiment with different criteria for selecting the next best attribute, and save/print the resulting decision tree for the best one.

12 **Exercise 8: Decision Tree Learning: On Scalability**

Using the `letter-recognition` database, which contains $N = 20,000$ examples, learn a decision tree for learning attribute 1 (lettr) from attributes 2 to 17 in the following ways:

2 1. Starting with only 100 examples, and increasing in chunks of 100, determine the maximum number n of examples the applet can handle in reasonable time (say, 15 minutes).

10 2. Learn and save/print the decision trees with sets containing integer multiples of 100. Describe the structural changes occurring, if any. Determine and compare the prediction rate for each letter on the training set and a test set containing all examples not used for training, and visualize the results in appropriate graphics.

30 **Exercise 9: Decision Tree Learning: The Effect of Noise**

Using the `letter-recognition` database, do the following:

2 1. Implement an algorithm which determines whether a data set contains noise.
(*Hint: First specify a unique criterion for the presence of noise!*)

3 2. Extend the algorithm such that it quantifies the amount of noise.
(*Hint: First find an appropriate measure and representation for the amount of noise!*)

5 3. Extend the algorithm such that you can modify the amount of noise.
(*Hint: It should be possible to specify for each letter class the amount of noise from another letter class!*)
(*Note: It should be possible to specify 0 as amount of noise!*)

2 4. Using these tools, determine whether the `letter-recognition` database contains noise. If yes, quantify the amount.

18 5. For this experiment, first choose arbitrarily a training set with the maximum number of examples the applet can handle, then chose another set of equal size for testing.
Create variants of the training and test data sets containing 1%, 5%, 10%, 20%, and 50% noise. Train decision trees and determine their performance on *all* training and test data sets. Analyze, visualize, and describe the results.