Compulsory Assignment 1 - INF 283

September 14, 2015

Goals of this exercise

- Learn to implement algorithms for decision tree learning.
- Learn to evaluate performance of machine learners with respect to practical applications.

1 ID3

- Implement ID3 in Python or Java and make necessary modifications to handle cases where some of the training examples are missing values for some attributes. Use the *GainRatio* as defined by Mitchell for selecting which attributes to use for tree expansions (Eq. 3.6 in Mitchell). You may use standard libraries in your chosen programming language, but make sure you provide your own implementation of *ID3*, *entropy* and *GainRatio*.
- Implement reduced-error pruning to generalise a trained tree with respect to a given validation set.
- Make a program that takes as arguments:
  - A comma-separated plain text file of categorical variables, to be used for training and reduced-error pruning of the decision tree.
  - A number specifying (using 0-based index) which column to use as target attribute.
  - A number in the range [0,1] specifying the fraction of the provided data that should be reserved for reduced-error pruning (the validation set).
  - *Optional*: If you want to support headers in the data file, add an argument to specify whether the first line should be interpreted as a header describing the variables corresponding to each column.

  The program should use pseudo-random sampling to construct a training set and a validation set in accordance with the input. Then it should compute the tree, apply reduced-error pruning, and produce as output:
  - Accuracy for its predictions for the training set, before and after pruning.

– Accuracy for its predictions for the validation set, before and after pruning.
– Sensitivity and specificity for its predictions for the validation set after pruning.
– A description of the tree, before and after pruning, as a set of rules, sorted by the length of the rules. Format the rules as `<attribute = value> AND <attribute = value> ... --> target attribute = value`.

For sensitivity and specificity, you should compute them considering each of the possible values of the target attribute as positive cases, while considering all other possible values as negative. Your program does not need to handle malformed input, and you may restrict the number of possible values for each attribute to those actually observed in the data set.

2 Application

- Download the mushroom data-set, referred in Lantz, chap. 5. Use the full version of the set, available from: `http://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/`. This data set contains categorical variables describing different kinds of mushroom and a classification for each mushroom as either *edible* or *poisonous*. A description of the data file can be found in `agaricus-lepiota.names`.
- Choose a fraction to use for validation, and run your program on the file `agaricus-lepiota.data`, to train a decision tree that will predict whether a mushroom is *edible* or *poisonous*.
- *Optional:* If your decision tree program support headers, you may add a descriptive header to the data file.

3 Interpretation

(a) Compare the accuracy of predictions for the training and validation data, before and after pruning. Comment on the results.

(b) Compare the rules derived from the tree before and after pruning. Can you formulate some rules of thumb for mushroom selection, in plain English?

(c) Assuming that the trained tree is going to be used to safeguard novice mushroom pickers from eating poisonous mushroom, which of the measures you calculated (accuracy, specificity or sensitivity) are more appropriate for evaluating the performance of the tree?

(d) Assume a mushroom picker have trusted the tree to identify x different kinds of mushrooms. Express an estimate for $p(x)$, the probability that at least one of them is poisonous.

(e) Considering the potentially severe consequences of eating a poisonous mushroom, it is not advisable to trust any inductive inference about the edibility of an unseen instance. Comment on practical situations where the rules derived from such a machine learner might still be useful.

Submission

Submit source code (Exercise 1) and plain text files or pdfs with your answers (Exercise 3) in the appropriate directory in *Studentportalen*. Submit the assignment by the 28th of September. Please answer the assignment in English.