Stony Brook University
Department of Computer Science
CSE 353: Machine Learning
(Spring 2014)
Instructor: Luis E. Ortiz

Programming Homework 2: Decision Trees

Out: *Mon Apr 14*, **Due**: *Mon Apr 21 at midnight (11:59pm)*

Space Shuttle Autolanding Controller using Decision Trees

In this homework you will implement the decision-tree machine-learning technique for classification and learn a decision tree to represent "comprehendible rules for determining the conditions under which an autolanding would be preferable to manual control of the spacecraft." [1] The dataset comes from the UCI Machine Learning Repository [Frank and Asuncion, 2010]. (Please visit `http://archive.ics.uci.edu/ml/datasets/Shuttle+Landing+Control` for more information on this particular data set.)

The binary target concept is the advise on using manual or automatic control (`AUTO`). The controller decision rule may consider six (6) attributes, labeled as `STABILITY`, `ERROR`, `SIGN`, `WIND`, `MAGNITUDE`, and `VISIBILITY`. The target concept and the attributes are categorical types, but one can also model them as either binary or ordinal types, with values (and integer mappings) given by the following table.

attributes	**values**
`STABILITY` (1)	`stab` (1), `xstab` (2)
`ERROR` (2)	`XL` (1), `LX` (2), `MM` (3), `SS` (4)
`SIGN` (3)	`pp` (1), `nn` (2)
`WIND` (4)	`head` (1), `tail` (2)
`MAGNITUDE` (5)	`Low` (1), `Medium` (2), `Strong` (3), `OutOfRange` (4)
`VISIBILITY` (6)	`yes` (1), `no` (2)
concept	**values**
`AUTO`	`noauto` (1), `auto` (2)

The original data set of examples consists of only 16 examples, including "don't care" conditions. The "don't care" conditions in some examples of the original data set have been expanded into all possible specific assignments yielding a data set of 253 complete specific examples. [2] Each line in the file `shuttle_ext_unique.dat` corresponds to each of those 253 examples. For instance, the following data example

```
AUTO=auto, STABILITY=xstab, ERROR=MM, SIGN=pp, WIND=head, MAGNITUDE=OutOfRange,
VISIBILITY=no
```

is represented in the data set file as the following space-separated list of seven (7) integers,

```
2  2  3  1  1  4  2
```

where the first integer corresponds to the target concept value (i.e., the output) and the last six integers correspond to the attribute values in order (i.e., the input).

[1] `http://archive.ics.uci.edu/ml/machine-learning-databases/shuttle-landing-control/shuttle-landing-control.names`

[2] Note that there are only 3 cases missing for the input data to cover the whole feature space.

Learning a Decision Tree for Space Shuttle Autolanding Controller

Implement the decision-tree machine-learning technique (see reading material on decision trees from Russell and Norvig's AI book for more information; available in the Document Section of Blackboard), using `noauto` as default, and apply it to the space shuttle autolanding controller data.

Perform the following steps:

1. Apply the decision-tree learning algoritm to the training data, using *maximum information gain*, or equivalently, *minimum average entropy*, as the measure to select attributes.
   (a) Draw the resulting tree.
2. [**Extra Credit**] Repeat the previous step, this time using *gain ratio* as the attribute-selection measure.
   (a) [**Extra Credit**] Compare and contrast the two resulting trees.

What to Turn In

You need to submit the following.

1. A **written report** (*in PDF*) that includes the drawing of the decision tree you found for the shuttle autolanding controller data. The internal nodes, branches and the leafs of the tree should be properly and clearly labeled with the corresponding attribute, attribute values and output/class. *For the extra-credit parts only,* a clear and properly labeled drawing of the decision tree learned based on gain-ration along with a brief statement comparing and contrasting the two trees.

   You can draw the decision tree(s) by hand and turn it in as a hard copy. For the extra-credit you can do the same, including your report with the additional information, of course.

2. All your **code and executable** (as a tared-and-gziped compressed file), with instructions on how to run your program. A platform-independent executable is preferred; otherwise, also provide instructions on how to compile your program. Please use standard tools/compilers/etc. generally available in most popular platforms.

Collaboration Policy: *It is OK to discuss the homework with your peers, but each student must write and turn in his/her own report, code, etc. based on his/her own work.*

References

A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.