Wan D. Bae - Data Science (ITE4005)
컴퓨터전공 2013012289 한기훈

April 20th, 2017

Report for Assignment2 - Decision Tree

1. Summery of algorithm.

The decision tree build algorithm I implemented follows the following procedure.

- Read the training data and test data.
- Build decision tree using C4.5 (using GainRatio) algorithm.
- Generate test results using decision tree.
- Write results into result file.

2. Detailed description of code.

This code has three major parts.

A. `getTrainingData(), getTestData()`

Read training data, test data from each given path. Since first line represents attribute name, I read first line separately. Also, given data file has carriage return '`\r`', so I have to remove it too.

Return type is `vector< vector< pair<string, string> > >`.

B. `processLearning()`

First, it converts training data structure from `vector< vector< pair<string, string> > >` to `vector< map<string, string> >`. It makes easy to handle data tuples. Then, it creates attribute name list "`set<string> attr_list`"

and value list for each attribute "`map<string, set<string> > orig_attr_list`".

Then, call `learn()` function to create each node recursively.

In `learn()` function, following procedures will be executed.

- Check if current data tuples are pure. If pure, return its class label.
- Check if attribute list is empty. If so, select class label using majority voting and return it.

- Select attribute using GainRatio and erase it from current attribute list.
- Generate each branch for each value belongs to selected attribute.
  - Create partial data tuples which matches to attribute value.
  - If created partial data tuples is empty, select class label using majority voting and return it.
  - Else, proceed to `learn()` function recursively with partial data tuples and assign returned node into branch.
- Return created node.

C. `createTestResult()`

First, convert test data structure and create attribute list like in `processLearning()` function. Then, get result of each test tuple using `testFunction()`.

in `testFunction()`, following procedures will be executed.

- If current node has no more branch, it means this node is leaf node. Therefore, return its name(=class label).
- Else, get attribute value from current test data tuple.
- If there is no branch for attribute value, get class label from each branch and selects majority of them.
- Otherwise, proceed to next node.

3. How to compile this code.

Just type ‘`make`’ in terminal. Or, please type below line.

```
$ g++ -O2 -o dt.exe main.cpp data_read.cpp learning.cpp testing.cpp --std=c++11
```

Above task will generate ‘`dt.exe`’ file into same directory.

Compiler must support C++11 standard.

4. Any other specification.

Has a problem with majority value selection. Tie breaking method needed.