AINT351 Machine Learning 2016/17
P2.1: Classification with Decision Trees

Part of your AINT351 laboratory journal coursework

This set of laboratory practical exercises are part of the AINT351 laboratory journal coursework. After you have completed the exercises you must write a corresponding lab report and include it in the final lab journal according to the instructions provided in the AINT351 Coursework specification document available from the module DLE site.

Important note on practical

This should describe:

- What the lab tasks were
- How you tackled them
- Your results and findings.

1. Initial Decision Tree

- Write a function, *learnDecisionTree*, that takes as parameters, a matrix of variable values and a vector of classifications, e.g., the *meas* matrix and *species* vector produced by loading the Fisher iris data set. The function should construct a data structure appropriate for representing an increasing number of data sets. The data sets will be produced by repeatedly splitting the initial data set according to the decision tree learning algorithm presented in class.
- This function should also call the functions below, initially to test them, but eventually to implement the full decision tree learning algorithm.

2. Split

- Write a function, *split*, that takes a set, a variable index and a threshold value, as parameters and returns two sets. The two sets should be the result of splitting the original set on the given variable and value.

3. Entropy

- Using the Fisher's iris data set, write a function *entropy* that takes a data set as an argument and returns the entropy of that data set.

4. Improvement

- Write an *improvement* function that takes three sets as paramters, one original set and two sets that have been constructed by splitting the original set. The function should caluculate the change in entropy from the original to the two new sets.

5. Max Split

- Write a *maxSplit* function that loops through allo possible sets, variables and values and calls the functions above in order to find the best possible next split given the current sets.

6. Final Decision Tree

- Finalise the *learnDecisionTree* function so that it keeps splitting up the original set until there are no further splits that improve the overall entropy. The function should now construct an array of splitting rules that what sets, variables and values have been used when ant what new sets were created. This set is the actual decision tree and it should be returned by the *learnDecisionTree* function.

7. Majority Class

- Write a *majorityClass* function that takes a set as a parameter and returns the name of the majority class within that set. Use this function inside your *learnDecisionTree* function so that it records the majority class for each new rule.

8. Classify

- Write a *classify* function that takes a decision tree and a sample data vector as parameters and returns the appropriate classification.