**CM0669 Machine Learning and Computer Vision**

**Lab 4** decision trees: classification/regression

**1. Classification with decision trees**

Download the matlab codes ‘Tree\_SL\_SW\_out\_SE\_VERS.m’, ‘Tree\_PL\_PW\_out\_SE\_VERS.m’, ‘Tree\_SL\_SW\_PL\_PW\_out\_SE\_VERS.m’, ‘Tree\_SL\_SW\_out\_VERS\_VIRG.m’, ‘Tree\_PL\_PW\_out\_VERS\_VIRG.m’, ‘Tree\_SL\_SW\_PL\_PW\_out\_VERS\_VIRG.m’ in a folder ‘Week4’.

Each Matlab code implements a classification tree. However, as you have seen in previous labs, they differ in terms of the input and output used. For instance, ‘Tree\_SL\_SW\_out\_SE\_VERS.m’ is a classifier using as input **S**epal **L**ength and **S**epal **W**idth and gives as output the ‘**SE**tosa’ and ‘**VERS**icolor’ class labels.

Similar to what you have seen in the previous Lab sessions for the evaluation of each classifier, 80 samples (specimens) have been used for the training while the testing is performed on 20 samples.

1. Open up Matlab and browse ‘help using the desktop’. A new window will open. Type in the ‘search results’ tab ‘classregtree’ (without quotation marks). A helpful description of the built-in function ‘classregtree’ will be given with good examples.
2. Open up each Matlab code using the Matlab workspace and ensure you understand its content (note that each Matlab code is well commented.).
3. If the classifier uses as input Petal length and Petal width and as output ‘setosa’ or ‘versicolor’, calculate the information gain for attribute Petal Length with a threshold equal to 2.6. Comment the results (run Tree\_PL\_PW\_out\_SE\_VERS.m before you calculate the information gain by hand.)
4. Run all the codes and complete the results in the table given below.
5. Compare and analyse the results.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Classification Tree | | | | | | | | |
| Input | | | | Output | | | Evaluation | |
| Sepal  Length | Sepal  Width | Petal  Length | Petal  Width | Setosa | Versicolor | Virginica | Training Error (%) | Testing  Error (%) |
| Yes | Yes | No | No | Yes | Yes | No | 0 | 0 |
| No | No | Yes | Yes | Yes | Yes | No |  |  |
| Yes | Yes | Yes | Yes | Yes | Yes | No |  |  |
| Yes | Yes | No | No | No | Yes | Yes |  |  |
| No | No | Yes | Yes | No | Yes | Yes |  |  |
| Yes | Yes | Yes | Yes | No | Yes | Yes |  |  |

**3. Regression tree**

Similar to what you have seen in the last week, the same training and testing sets will be used to evaluate the performance of a regression tree. Refer to the last part of Lab 3 for more details on the regression problem.

The Matlab programme ‘Tree\_regression.m’ is an implementation of a regression system using a decision tree.

1. Open up the Matlab code and ensure you understand its content (note that the Matlab code is well commented.).
2. Run the code and complete the results in the table given below.

|  |  |  |  |
| --- | --- | --- | --- |
| Temperature  (degrees Celcius) | Length (mm) | Actual age  (days) | Estimated age with regression tree |
| 25 | 394 |  |  |
| 25 | 447 |  |  |
| 27 | 326 |  |  |
| 27 | 402 |  |  |
| 29 | 214 |  |  |
| 29 | 289 |  |  |
| 31 | 271 |  |  |
| 31 | 302 |  |  |

1. Add a piece of Matlab code to calculate both the training and the testing regression errors which are given by:

Training\_error = ;

for all training samples *i=1, 2, ..., Ntraining.* (*Ntraining*=36).

Testing\_error = ;

for all test samples *i=1, 2, ..., Ntesting.* (*Ntesting*=8).

1. Complete this table and interpret the results:

|  |  |
| --- | --- |
| Training error |  |
| Testing error |  |