University of Central Florida

CAP6676

Knowledge Representation

Submitted To:
Guo-Jun Qi
Asst. Professor
Computer Science
Department

Submitted By :
Pooya Abolghasemi
Fall 2015

Contents

1 Step 1: Choosing the dataset

I chose the Statlog(Heart) dataset. This database contains 13 attributes (which have been extracted from a larger set of 75). It contains different types of attributes(Real, Binary, Nominal and Ordered). The 13 features are:

1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholesterol in mg/dl
6. fasting blood sugar more than 120 mg/dl
7. resting electrocardiograph results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 3 = normal; 6 = fixed defect; 7 = reversible defect

The dataset contains 270 observations and no missing values. The data types as recorded in the Machine Learning Repository is multivariate. The default task is classification and attribute types are categorical and real.

2 Step 2: Split the dataset into training and test sets

For the k-fold process I chose k equal 5. As you can see the Python code below in each iteration one of the fold is selected for test set and the rest for training the tree. After training is completed each observation in the test set will be the input to the predict function of the tree. The predict function will find the decision tree prediction for the observation and at last we can compare the results with the real observation from the dataset.

```
k_fold=5
fold_size=kfold(data,k_fold)
test_result=[]
for i in range(k_fold):
        test_data=data[int(i*fold_size):int((i+1)*fold_size),:]
    train_data=
        np.delete(data, range(int(i*fold_size),int((i+1)*fold_size)), 0)
    decision_tree=train(train_data,attributes)
    for toTest in test_data:
        test_result.append(decision_tree.predict(toTest))
```

3 Step 3: Implementing the decision tree

I implement the decision tree in Python. I trained the recursively. At each iteration the best attribute for splitting the data(based on the gain calculated for the attribute) will be selected. Then data will divided into groups based on different options of the selected feature and will be passed to the same function to do same procedure recursively. Here is the pseudo code for the train function.

```
def train(data, attributes):
        selected=find_the_best_attribute(data)
    node new_node=create_node(selected)
    if entropy(data)>threshold:
      data_divided=[]
      data_divided=
        new_node.split_data_for_selected_attribute(data,selected)
      for child_data in data_divided
          child_node=train(child_data,attributes)
          new_node.add_child(child_node)
    return new_node
```

Attributes can be selected more than one time in the tree. Best attribute is selected by extracting the best splitting gain for the data. The stop criteria is the entropy in a node. If Entropy for node become lower than the defined threshold the algorithm consider that node a leaf node and does not expand the node anymore.

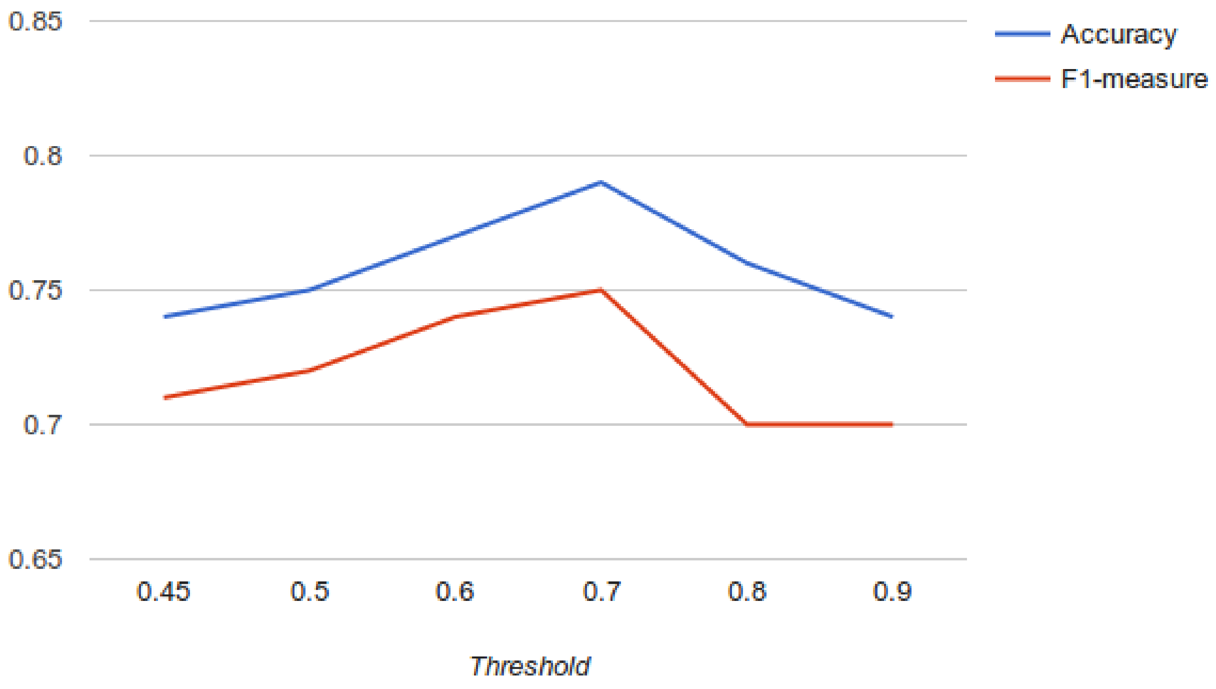
4 Step 4: Report the Performance

Here is the Accuracy,Precision,Recall and F1-measure for training the tree over 4 folds and testing with the remaining one.

Test Fold Index	Accuracy	Precision	Recall	F1-Measure
1	72.2%	73.6%	58.3%	0.65
2	76.3%	71.4%	66.6%	0.68
3	70.5%	78.5%	61.1%	0.68
4	63.8%	50%	53.8%	0.51
5	71.8%	73.3%	68.7%	0.7
Average	70.92%	69.36%	61.7%	0.64
Standard Deviation	4.53	11.13	6.06	0.07

5 Step 5: Over Fitting of Decision Tree

For investigating the over fitting, I tested the algorithm with several threshold from 0.45 to 0.9. As You can see in the graph for accuracy and F1-measure, the test error is minimum when threshold is 0.7. When the threshold is more than 0.7 the tree has a larger depth and it is over fitted to the data. However, when the threshold is a large number closer to 1 the tree is under fitted meaning that it didn't expand enough.


Threshold
Accuracy
F1-measure