Assignment Report 3 - Machine Learning COL774

Piyush Kaul - 2015EEY7544

April 10, 2016

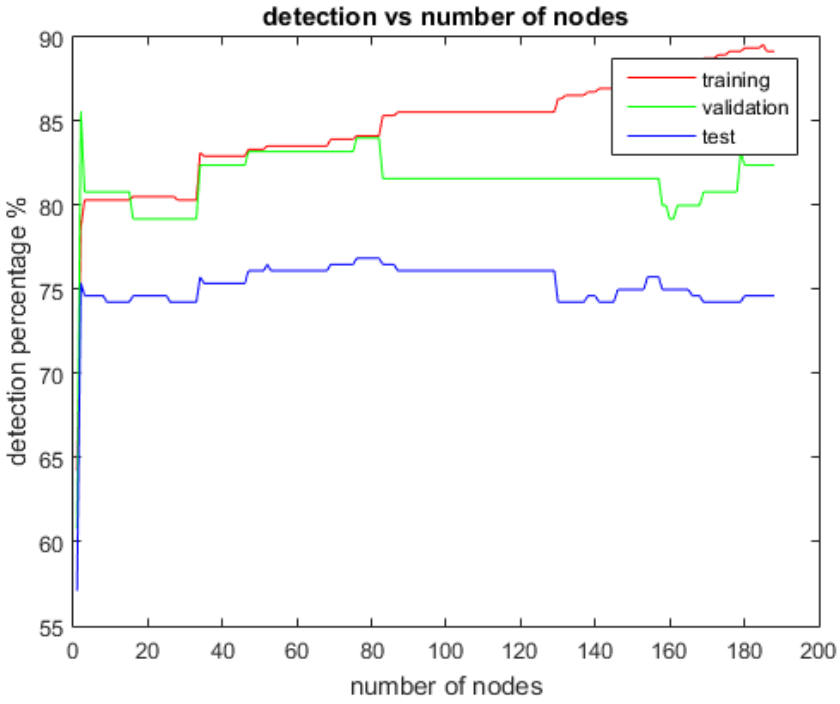
1 ANS-1 Decision Tree for Classificatin

1.1 PART A

Q. Construct a decision tree using the given data to predict whether a passenger survives the incident. The dataset contains a mix of categorical as well as numerical attributes. Preprocess (before growing the tree) each numerical attribute into a Boolean attribute by a) computing the median value of the attribute in the training data (make sure not to ignore the duplicates) b) replacing each numerical value by a 1/0 value based on whether the value is greater than the median threshold or not. Note that this process should be repeated for each attribute independently. For non-Boolean (discrete valued) attributes, you should use a multi-way split as discussed in class. Use information gain as the criterion for choosing the attribute to split on. In case of a tie, choose the attribute which appears first in the ordering as given in the training data. Plot the train, validation and test set accuracies against the number of nodes in the tree as you grow the tree. On X-axis you should plot the number of nodes in the tree and Y-axis should represent the accuracy. Comment on your observations

Ans. The following is the plot of training, validation and test detection percentage vs the number of nodes in the tree.

Figure 1.1: Change in Training, Validation and Test Detection percentage as the number of nodes are added to the decision tree



1. As can be noticed from the plot above, as nodes are added the detection percentage for the training set continues to increase. This is as per expectation as extra nodes represent the training data more exactly.

2. The validation percentage initially increases as nodes are added and later decrease as more nodes are added causing overfitting.

3. The test percentage also initially increases as nodes are added and later decrease as more nodes are added causing overfitting.

Final Results are as follows

Training	89.1566%
Validation	82.4000%
Test	74.6269%

1.2 PART B

Q. One of the ways to reduce overfitting in decision trees is to grow the tree fully and then use post-pruning based on a validation set. In post-pruning, we greedily prune the nodes of the tree (and sub-tree below them) by iteratively picking a node to prune so that resultant tree gives maximum increase in accuracy on the validation set. In other words, among all the nodes in the tree, we prune the node such that pruning it(and sub-tree below it) results in maximum increase in accuracy over the validation set. This is repeated until any further pruning leads to decrease in accuracy over the validation set. Post prune the tree obtained in step (a) above using the validation set. Again plot the training, validation and test set accuracies against the number of nodes in the tree as you successively prune the tree. Comment on your findings. We have posted the relevant sections from Mitchell's textbook for this part on the course website (aceessible only through an internal link inside IIT)..

Ans. As shown in the figure below, the Validation error decreases as we iterate over the nodes and prune those which meet the criteria of decrease in validation error.Also the test error end up decreasing as well. However the training error increases by pruning nodes.

The final detection percentage results are as follows

	Before Pruning	After Pruning
Training	89.1566%	78.7149%
Validation	82.4000%	85.6000%
Test	74.6269%	75.3731%

Figure 1.2: Change in Training, Validation and Test Detection percentage as we iterate over nodes and prune those which decrease validation error

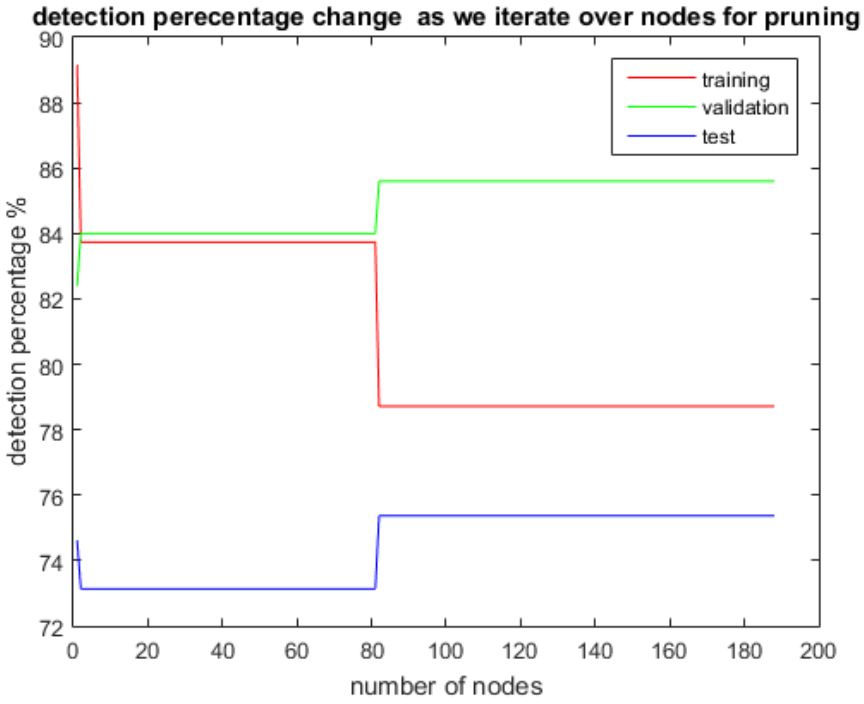
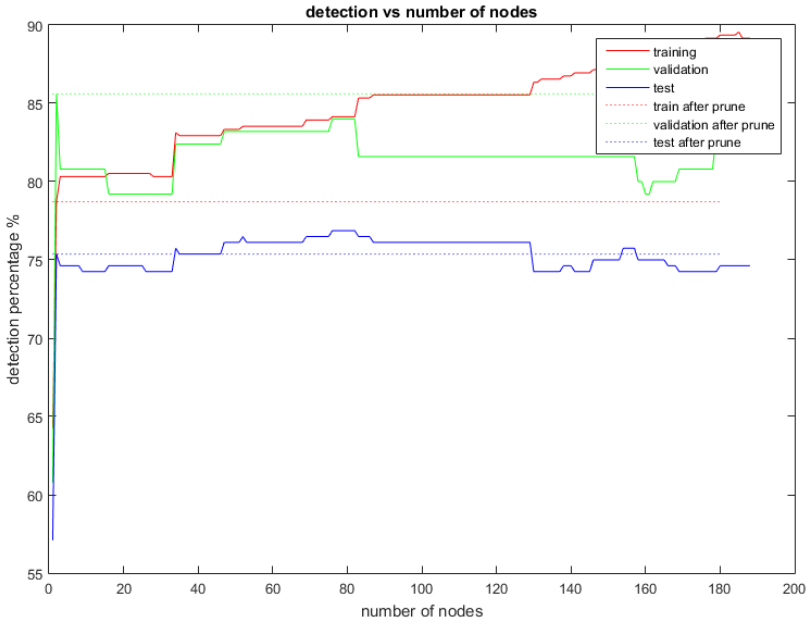


Figure 1.3: Final Tree After Pruning Compared to Original Tree



1.3 PART C

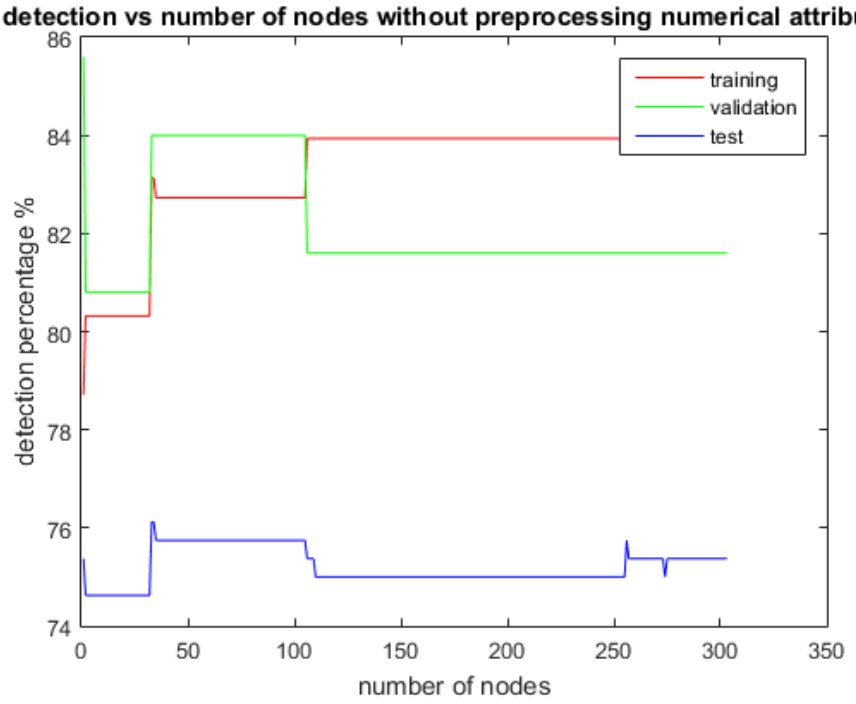
Q. In Part(a) we used the median value of a numerical attribute to convert it in a 1/0 valued attribute as a pre-processing step. In a more sophisticated setting, no such pre-processing is done in the beginning. At any given internal node of the tree, a numerical attribute is considered for a two way split by calculating the median attribute value from the data instances coming to that node, and then computing the information gain if the data was split based on whether the numerical value of the attribute is greater than the median or not. As earlier, the node is split on the attribute which maximizes the information gain. Note that in this setting, the original value of a numerical attribute remains intact, and a numerical attribute can be considered for splitting in the tree mutliple times. Implement this new way of handing numerical attributes. Report the numerical attributes which are split multiple times in a branch (along with the maximum number of times they are split in any given branch and the corresponding thresholds). Replot the curves in part (a). Comment on which results (Part (a) or Part (c)) are better and why. You don't have to implement pruning for this part

Ans. The final results for decision tree without pre-processing are as follows

Training	83.734940%
Validation	81.600000%
Test	75.373134%

So the test results are somewhat better than the unpruned tree with preprocessing

Figure 1.4: Change in Detection percentage with addition of nodes in Decision Tree without Preprocessing



The following are the maximum times an attribute is split on any particular branch. The attributes used multiple times are marked in **bold**

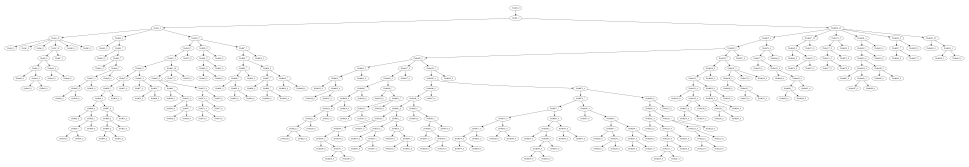
Attribute	Split number of times
Pclass	1
Sex	1
Age	**6**
SibSp	**2**
Parch	1
Ticket	**4**
Fare	**3**
Embarked	1
`Cabin_a`	1
`Cabin_b`	1

The Attribute "Age" is split 6 times in the same branch. The thresholds used for various splits are.

Split Order	Median value of Age
1st split	22
2nd split	32
3rd split	44
4th split	38
5th split	42
6th split	39

Final Tree

Figure 1.5: Final Tree



1.4 PART D

Q.There are various libraries available for decision tree implmentation in Python, Matlab etc. Use the scikit-learn library of Python to grow a decision tree. Click here to read the documentation of the library and various parameter options to grow the decision tree. Try growing different trees by varying parameters of the learning algorithm. Some exampes of the parameters that you can play with include min samples split, min samples leaf and max depth (feel free to vary any other paramters as well). Find the setting of parameters which gives you best accuracy on the validation set. Report the parameter setting giving you best validation set accuracy as well as the validation and test set accuracies for this setting. Explain briefly why the particular parameter setting you obtained works best

Ans. The best accuracy with scikit-learn library and validation set was obtained for the following parameters

Parameter	Value
`min_samples_split`	8
`min_samples_leaf`	3
`max_depth`	8

The accuracy obtained are as follows.

Parameter	Value
training accuracy	0.91%
validation accuracy	0.82%
test accuracy	0.79%

The reason why these parameters work is keeping a minimum number of samples per split and minimum samples per leaf ensures that the tree representation and splits are not based on too few samples of data. So any noise or training data specific behavior is not represented in the decision tree and generalization of the tree is preserved.

By limiting the depth of the tree also, we try to keep the generalization adequate.

Limits on all three attributes essentially help in avoiding overfitting, while allowing enough paramters to avoid underfitting as well.

2 ANS-2 NAIVE BAYES

Q. In this problem, we will use the naive Bayes algorithm to learn a model for classifying an article into a given set of newsgroups. The data and its description is available through the UCI data repository. The original data is description available here. We have processed the data further to remove punctuation symbols, stopwords etc. The processed dataset contains the subset of the articles in the newsgroups rec.* and talk.*. This corresponds to a total of 7230 articles in 8 different newsgroups. The processed data is made available to you in a single lile with each row representing one article. Each row contains the information about the class of an article followed by the list of words appearing in the article.

2.1 PART A

Q. Implement the Naive Bayes algorithm to classify each of the articles into one of the newsgroup categories. Randomly divide your data into 5 equal sized splits of size 1446 each. Make sure NOT TO create the splits in any particular order (e.g. picking documents sequentially) otherwise it may lead to skewed distribution of classes across the splits. Now, perform 5-fold cross validation (train on each possible combination of 4 splits and the test on the remaining one). Report average test set accuracies. Notes:

- Make sure to use the Laplace smoothing for Naive Bayes (as discussed in class) to avoid any zero probabilities.
- You should implement your algorithm using logarithms to avoid under ow issues.
- You should implement naive Bayes from the first principles and not use any existing Matlab modules

Ans.
Following results were obtained for five-fold crossvalidation

Cross-Validation Set	Detection Percentage
1	89.419087%
2	88.450899%
3	87.551867%
4	88.589212%
5	89.764869%
Mean	**88.755187**%
Standard Deviation	**0.0087**

2.2 PART B

Q. What is the accuracy that you would obtain by randomly guessing one of the newsgroups as the target class for each of the articles. How much improvement does your algorithm give over a random prediction?

Ans. The expected accuracy for randomly guessing over 8 equiprobable newsgroups is 1/8=12.5%. The improvement obtained is 88.75% - 12.5% = 76%

2.3 PART C

Q. Some (4% in the original data) of the articles were cross-posted i.e. posted on more than one newsgroup. Does it create a problem for the naive Bayes learner? Explain
Ans. The naive Bayes Learner relies on the difference in probability of occurrences of specific words across newsgroups. If the same words get posted across multiple newsgroups, the probabilities of these words will not be as useful to discriminate across newsgroups. However since the proportion is small (4%), the Naive Bayes Learner should still work fairly reliably.

2.4 PART D

Q. For each of the train/test splits in part (a) above, vary the number of articles used in training from 1000 to 5784 (training split size) at the interval of 1000 and evaluate on the test split (sized 1446). Plot on a graph the train as well as the test set accuracies (y-axis) averaged over the 5 random splits, as you vary the number of training examples (x-axis). This is called the learning curve. What do you observe? Explain

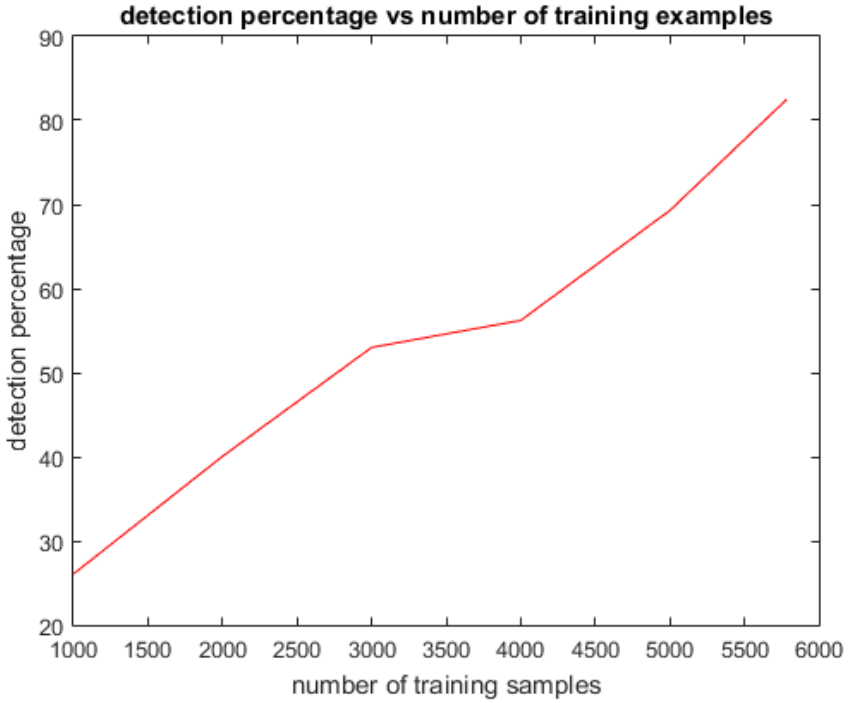
Ans.

Following results were obtained for five-fold crossvalidation

Cross-Validation Set	1	2	3	4	5	Mean
1000	25.4495%	25.6570%	25.7953%	25.0346%	28.3541%	**26.0581**%
2000	39.1425%	39.8340%	38.9350%	39.8340%	42.5311%	**40.0553**%
3000	52.7663%	53.1812%	51.3831%	52.2822%	55.3942%	**53.0014**%
4000	55.7400%	55.1176%	65.5602%	50.5533%	54.0802%	**56.2103**%
5000	70.6086%	69.5021%	70.5394%	66.5975%	69.1563%	**69.2808**%
5784	82.9876%	84.0941%	84.3707%	80.9820%	80.0138%	**82.4896**%

Here is the plot of the above tabulated data.

Figure 2.1: Change in Detection Percentage as we increase the number of training samples



2.5 PART E

Q. Read about the confusion matrix. Draw the confusion matrix for your results in the part (a) above. Which newsgroup has the highest value of the diagonal entry in the confusion matrix? Which two newsgroups are confused the most with each other i.e. which is the highest entry amongst the non-diagonal entries in the confusion matrix? Explain your observations. Include the confusion matrix in your submission.
Ans. The confusion matrix is shown below.

Figure 2.2: Confusion Matrix

CONFUSION MATRIX	rec.autos	rec.motorcycles	rec.sport.baseball	rec.sport.hockey	talk.politics.guns	talk.politics.mideast	talk.politics.misc	talk.religion.misc
rec.autos	129	4	0	0	0	0	0	0
rec.motorcycles	1	140	0	0	0	0	0	0
rec.sport.baseball	0	0	158	0	0	0	0	0
rec.sport.hockey	0	0	6	186	0	0	0	0
talk.politics.guns	2	2	2	0	142	0	3	2
talk.politics.mideast	14	9	6	2	7	186	5	6
talk.politics.misc	52	44	27	11	32	2	146	12
talk.religion.misc	0	0	1	0	0	0	1	106

1. The maximum values on the diagonal are marked in blue. Those correspond the Newsgroups **rec.sport.hockey** and **talk.politics.mideast**. i.e. these groups are most unlikely to be confused with other groups. The reason for this is that these groups have lesser

overlap in specialized words used with other groups.

2. The maximum values off the diagonal is marked in red. This corresponds to the Newsgroups **talk.politics.misc** and **rec.autos** i.e. these groups are most likely to be confused with each other. A lot of common words would be getting used between these Newsgroups.