Decision Tree Classification

(Supervised Learning)

Introduction:

In Computer Science, data structures are used to organise and analyse the data efficiently. Each data structure can implement one or more Abstract data types. Arrays, linked lists, classes etc., are a few data structures. A different data structure is used to solve different problems. One such data structure is a tree. A tree has a root value with subtrees of children in a hierarchical structure. A tree with at most two child nodes is called a Binary tree.

Decision Tree:

A decision tree is the visualization of a complex decision making situation in which possible decisions and their outcomes are organized in the form of a graph that resembles a tree. A decision tree algorithm makes use of the binary tree that takes a decision to classify the given data set based on a attribute at each level in the decision tree.

Each rule or a data set can be classified in a decision tree is represented by tracing the path from root node to the next node and the next node and so on until ac action is reached.

A decision tree usually consists of :

1. Structuring the problem as a tree
2. Assigning subject probabilities to each event on the tree
3. Analysing the consequences
4. Identifying and selecting appropriate course of action based on the analysis

A decision tree algorithm is used to train a machine to make decisions at a given time based on the previous states the system has gone through. At every branch, there are a set of possibilities with a large set of attributes and using them we try to predict the next set of occurrences and decide the next path for traversing the decision tree. There are many attributes using which the path at a particular branch can be decided. For example, Gini index, Information Gain etc., If the information gain of an attribute is the highest among all other attributes, we use that attribute to perform the split. Hence, Information gain is calculated at every step for the whole system as well as for each attribute to build the whole decision tree.

Rahul Rachapalli 800968032

Data Sets:

A dataset is a collection of data represented in as a single statistical matrix. Given are two data sets

1. *Amazon reviews - Sentiment Analysis*
2. *Optical recognition of handwritten digits*

1. Amazon reviews sentiment analysis dataset

This dataset consists of reviews of Amazon products which is a subset of large Amazon review collection. The data set has 3 columns and the type of data is as follows:

Name of the product	*-*	*String / text*
Review of the product	*-*	*String / text*
Rating given by the user	*-*	*Integer set {1,2,3,4,5}*

2. Optical recognition of handwritten digits

This data set consists of preprocessed normalized bitmaps of handwritten digits from a preprinted form. From a total of 43 people, 30 contributed to the training set and different 13 to the test set. 32x32 bitmaps are divided into non-overlapping blocks of 4x4 and the number of on pixels are counted in each block. This generates an input matrix of 8x8 where each element is an integer in the range 0..16. This reduces dimensionality and gives invariance to small distortions.

The dataset has 65 columns with 64 columns being the 8X8 matrix integer values and the last column being the digit that the 64 values represented.

Amazon Dataset Analysis and Prediction

Assumptions:

The name of the product is ignored as it is independent of the rating and ratings cannot be predicted using the name of the product. Hence, the rating depends on the reviews only.

Preprocessing:

Step 1: The reviews are separated and each review is converted into an array of words.

Step 2: Each word in the array is compared with stop words that are defined. If the word is present in the list of stop words, it is ignored and we proceed to the next word.

Step 3: If the word is not present in the list of stop words, we lemmatize the word and then stem it. At the end of step 3, we get a list of words that are lemmatized and stemmed.

Step 4: The words that are lemmatized and stemmed are converted into a single string and the word frequency is counted.

Step 5: The top 500 words or the whole list of words are considered as the attributes to dive the data.

Step 6: A sparse matrix is built based on the above attributes counting the occurrences of attributes in each review.

Stop Words:

The words 'the','a','an,'is','it' etc., are present in almost all the sentences and do not contribute to the decision process. All such words are called stop words and it is important to remove the stop words in pre-processing stage. So, we used some libraries like nltk to get a list of some pre-defined stop words.

Lemmatizing[2]:

The process of grouping together different forms of word to analyse it as a single word.

Stemming[3]:

The process of converting words into their stem or its root form. For sing, sang, sung, singing are represented in their base word form like sing*

Decision Tree[1]:

With the sparse matrix built in the pre-processing stage and the ratings given by the user, the decision tree to predict the rating of a review is built.

Prediction :

The rating of a review is predicted after the new review is pre-processed (lemmed and stemmed) and a sparse matrix is built for it using the same set of attributes that we decided to build the decision tree. The new sparse matrix is given as input to the decision tree and the result is the predicted rating of the given review.

Results:

The experiment is conducted with different number of attributes that are considered after the pre-processing stage. The prediction rates are as follows:

Number of Attributes	Prediction Percentage
100	80.70
250	83.76
300	83.96
350	84.41
400	84.66
450	84.79
500	84.98
550	85.02
600	85.78
650	85.94
750	HPC job was killed as the job used a lot of memory
1000	HPC job was killed as the job used a lot of memory

Based on the above results, 500 attributes to build the decision tree is considered.

Pruning:

Pruning is the process of reducing the size of the decision tree by removing sections of the tree that does not provide any help to classify the given data sets.

The attributes considered to prune the decision tree in our case are `"min_samples_split"` , `"max_depth"` and `"min_samples_leaf"`

After building the decision tree for a sample set of the given training data, we found that 5 is an ideal value for **min_samples_split**

Prediction rates for different values of **min_samples_split** are:

min_samples_split	Prediction Rate(%)
15	77.5
12	77.5
10	79.17
8	80.83
7	82.5
6	82.5
5	83.33
4	85.83
3	86.67
2	93.33

The Prediction rates with `min_samples_split =5` and varying the **max_depth:**

max_depth	Prediction Rate(%)
10	78.33
15	80.83
16	81.67
17	83.33
19	83.33
20	83.33
22	83.33
24	83.33

Based on the above results, it is identified that the max_depth is not affecting the performance after reaching a certain level in the tree. Hence, I haven't imposed a boundary for the **max_depth** ie., the tree continues to grow until all the entries in the data set are classified correctly.

`Min_samples_leaf` implies the minimum number of samples that have to be present at each leaf node. I chose not to hard code any value for this as I didn't get to play around with this much.

To build a tree, I finally choose `max_depth` to be undefined and `min_samples_split` to **5.**

Other Methods:

We tried to calculate the polarity and subjectivity of each word in a sentence and I gave up soon as I had no idea about those.So, I chose a simple way to predict the rating of a review using word frequencies and build a sparse matrix out of it.

I also tried to use `TfidfVectorizer` to build the sparse matrix of the document and then use attributes from that matrix.

Visualization:

The decision tree is visualized using the Scikit learn library methods. One of the functions we used is `export_graphviz`. This function writes the decision tree in the form of text to the file specified. To visualize the decision tree, we used a tool, Web GraphViz. The text is input in the text field provided and the tree appears in the lower half of the a.ge

Graph output for Digit Recognition is in the file.

Digit Recognition Dataset Analysis and Prediction

Assumptions:

There are no missing values in the dataset and the given data is accurate.

Preprocessing:

The data is arranged in the form of a list, read from the file and is represented as a Data Frame.

Decision Tree :

The decision tree is built from the data frame that is obtained from the pre-processing step. This data frame itself acts as a matrix to build the decision tree.

Prediction :

A new dataset can be predicted using the decision tree built. The only step that is needed for it to be predicted is that there are **64** columns in the given integer data set. The pre-processing stage allows us to represent this data in the form of a list. The new list is input to the decision tree for prediction.

Results:

There are 64 columns and all of them are used as attributes to predict the results. The prediction probability for this dataset is **85.086%.**

Visualization:

The decision tree is visualized using the Scikit learn library methods. One of the functions we used is `export_graphviz`. This function writes the decision tree in the form of text to the file specified. To visualize the decision tree, we used a tool, Web GraphViz. The text is input in the text field provided and the tree appears in the lower half of the pag

Graph output for Digit Recognition is in thefile.

Data exploration Questions:

A. What is the number of attributes in each dataset?

Amazon reviews sentiment analysis dataset:

Number of Attributes : 500

Optical recognition of handwritten digits

Number of Attributes : 64

B. What is the number of observations?

Amazon reviews sentiment analysis dataset:

Number of Observations : 146824

Optical recognition of handwritten digits:

Number of Observations : 3823

C. What is the mean and standard deviation of each attribute?

Amazon reviews sentiment analysis dataset:

The attributes differ for each data set and I haven't calculated the mean and standard deviation of each of them in the sparse matrix generated.

Optical recognition of handwritten digits:

The mean and standard deviations are included in the attached files.

Testing DataSet

Training DataSet

D. What is the distribution of the different classes in each of the datasets?

Amazon reviews sentiment analysis dataset

Training Data Set:

Total: 146824

Class	Count	Distribution Rate(%)
1 star Rated Reviews	12146	8.72
2 star Rated Reviews	9040	6.16
3 star Rated Reviews	13364	9.10
4 star Rated Reviews	26509	18.05
5 star Rated Reviews	85765	58.41

Test Dataset:

Total :36707

Class	Count	Distribution Rate (%)
1 star Rated Reviews	3037	8.27
2 star Rated Reviews	2270	6.18
3 star Rated Reviews	3415	9.30
4 star Rated Reviews	6696	18.24
5 star Rated Reviews	21289	58.00

Optical recognition of handwritten digits

Training Data Set:

Total Training Data : 3823

Class	Training Data Count	Distribution Rate(%)
Digit 0	376	9.84
Digit 1	389	10.18
Digit 2	380	9.94
Digit 3	389	10.18
Digit 4	387	10.12
Digit 5	376	9.84
Digit 6	377	9.86
Digit 7	387	10.12
Digit 8	380	9.94
Digit 9	382	9.99

Test Dataset:

Total Testing Data : 1797

Class	Training Data Count	Distribution Rate(%)
Digit 0	178	9.90
Digit 1	182	10.12
Digit 2	177	9.84
Digit 3	183	10.18
Digit 4	181	10.07
Digit 5	182	10.12
Digit 6	181	10.07
Digit 7	179	9.96
Digit 8	174	9.68
Digit 9	180	10.02

Acknowledgement:

This assignment is a joint work with Manasa Sadanda, Vikas Deshpande, Adithya Vijaya Kumar and Sujal Vijayaraghavan.

References:

1) https://en.wikipedia.org/wiki/Decision_tree
2) https://en.wikipedia.org/wiki/Lemmatisation
3) https://en.wikipedia.org/wiki/Stemming
4) ITCS6156_SLProject.pdf
5) http://documents.software.dell.com/Statistics/Textbook/Text-Mining
6) http://scikit-learn.org/stable/modules/tree.html#tree