Department of Electrical & Computer Engineering

Pattern Recognition
SYDE-675
Assignment – 2

Prepared By:

Name: Kunal Taneja

Student ID: 20790967

Winter 2019

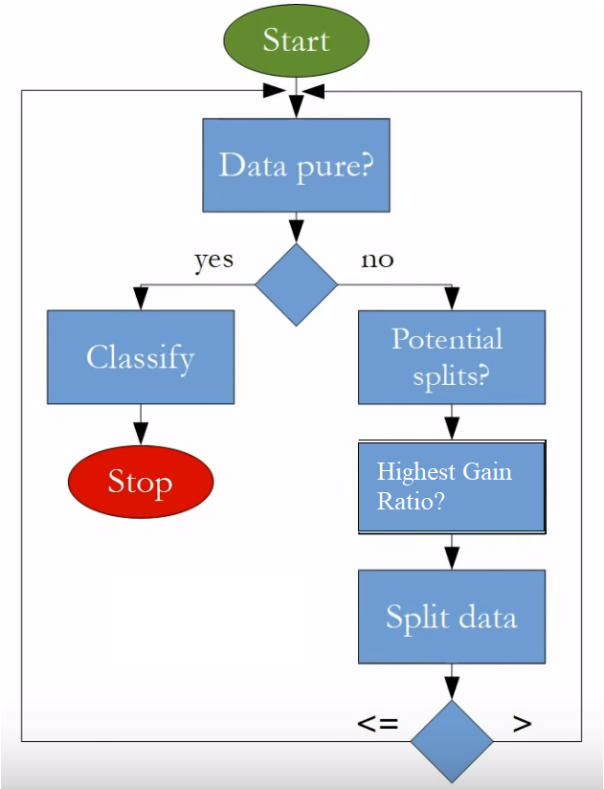
Question 1

In question 1 we have two datasets namely glass dataset (continuous) and tic-tac-toe dataset(categorical) and we need to implement C4.5 decision tree model over both these datasets.

Basically, C4.5 converts the trained trees (i.e. the output of the ID3 algorithm) into sets of if-then rules. The accuracy of each rule is then evaluated to determine the order in which they should be applied. Pruning is done by removing a rule's precondition if the accuracy of the rule improves without it.

We define an object-oriented based class structure for tree generation part and then the rules are evaluated from the tree. For continuous data we tend to find the threshold values where a split can be made over the feature with highest gain ratio. Whereas for the categorical data we do not need any kind of threshold value as split is made on the category value of the feature.

Flowchart of C4.5 (tree generation part)



Tree Pruning Part

After all the rules are fetched they are evaluated for their importance (all conditions within a rule are checked) with corresponding validation sets which are brought into picture by using 10 times 10-fold cross validation where training fold is further split into 20% validation sets. Rules are pruned if they have less informative preconditions based on the validation set accuracies. After evaluating the rules, they are ordered in decreasing order of their accuracies. Finally, the model is tested by feeding the unobserved test data. Because we need to implement 10 times 10-fold cross validation, we will get a total of 100 accuracies for 100 test sets.

In the end we get mean accuracy and variance over all accuracy values as follows:

DATASET	MEAN ACCURACY (fractional)	VARIANCE (fractional)
Glass - Continuous	0.6383549783549782	0.0116169351211559
Tic-Tac-Toe - Categorical	0.8314385964912281	0.0018582587836521

Question 2

Now, in this Question we add two types of noise that is:

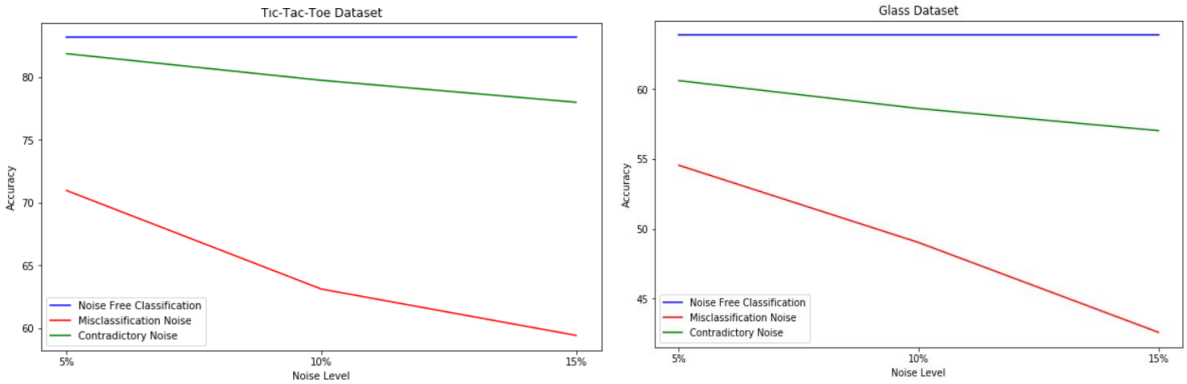
1. Contradictory Noise. The same sample appears more than once and is labelled with a different classification.

2. Misclassified Noise. A sample is labelled with the wrong class. This type of error is common in situations where different classes of data have similar symptoms.

After introducing these noises in our training sets we check the stability of our C4.5 decision tree model. Here again we perform 10 times 10-fold cross validation to calculate the accuracy of the noisy and noise free data. Following are the results of experimentation performed:

DATASET with noise 1	Accuracy with Noise 0%	Accuracy with Noise 5%	Accuracy with Noise 10%	Accuracy with Noise 15%
Glass	0.638354	0.605963	0.586020	0.570187
Tic-Tac-Toe	0.831438	0.818619	0.797410	0.779910

DATASET With noise 2	Accuracy with Noise 0%	Accuracy with Noise 5%	Accuracy with Noise 10%	Accuracy with Noise 15%
Glass	0.638354	0.545422	0.490187	0.425963
Tic-Tac-Toe	0.831438	0.709615	0.631115	0.594217



As we can see from the results obtained, contradictory noise doesn't penalise the decision tree algorithm that much and C4.5 model is somewhat stable towards contradictory noise. However, misclassified noise highly penalises the accuracy for the decision tree as accuracy values drop down drastically. Hence C4.5 is not stable towards misclassified noise.

Question 3

In this problem we use MNIST dataset, but only first 10000 samples in order to make computations fast. The first part of the problem asks us to implement a bidirectional search for feature selection algorithm to select top 10,50,150,392 sets of features. The objective function of bidirectional search is inter-class distance between the features.

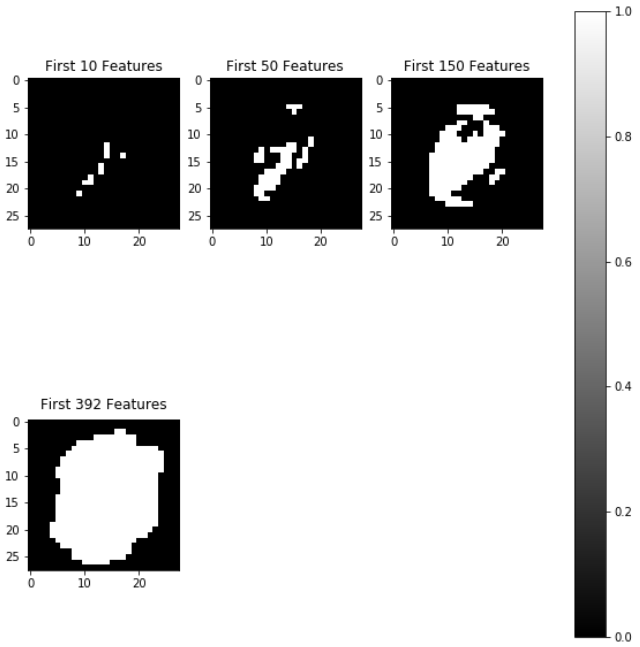
The best feature corresponds to one which has maximum interclass distance. For bidirectional search to happen a forward and backward feature list is maintained. Mutual exclusion is maintained for forward and backward searches which happen in parallel so that no same feature is added in forward list while it was removed from backward list and no feature is removed from backward list if it was added to forward list.

Finally, after selecting best 10,50,150,392 sets of features, kNN is applied (on test set of MNIST) over these set of features with k value of 3 and the following accuracy is obtained:

	Accuracy
10	0.5791
50	0.8109
150	0.9325
392	0.9490

As we can observe from the measurements with greater number of features we get better accuracies.

Now we plot these features on zero 2D plane with selected feature having value 1 and rest every feature value as 0. We get something as follows:



As we can see from the above figure, features or pixels on the corners are of least significance and hence are not selected in top 392 features. Features which represent the middle of the 28x28 image have highest interclass distance and are selected as best features by the bidirectional search. Whereas, pixels on the corners are padded on the MNIST digits and hence have no information towards digit classes.
Top 392 features represent a generic frame for all digit images present in the data.

At the end, we apply LDA over the MNIST dataset to bring the feature space into C-1 dimensions where C is the number of classes of the data. So, for this dataset, feature space is brought from 784 dimensions to 9 dimensions. In order to apply LDA we find scatter within and scatter betweenness of the datapoints. From these metrics we calculate the eigenvalues and eigenvectors. We see that we get only 9 eigenvalues having values greater than zero as rest of them have values nearly equal to zero. We then transform the feature space along the eigenvectors corresponding to highest eigenvalues. Now the dimension of our data is reduced to 10000x9
After applying LDA again we evaluate accuracies by using kNN model with k values as 3. Following are the results obtained.

	Accuracy
LDA_1	0.3097
LDA_2	0.5336
LDA_3	0.6692
LDA_4	0.7804
LDA_5	0.7926
LDA_6	0.8306
LDA_7	0.8516
LDA_8	0.8646
LDA_9	0.8745

So, the accuracy we obtain for 9 LDA components (features) is 87.45% which is not bad when compared to accuracy we get with 392 features that is 94.9%. It is far better from the accuracy we get for 10 features from part(a) which was 57.91%. So LDA tries to bring down the dimension of the data while preserving the classification properties of the data such that the data in lower dimension can be easily classified.