Machine Learning Project

Sem 2: 2023

29.04.2023

Code Link: https://colab.research.google.com/drive/1FaA0sQSi89Do-_cRiBsp0FqTD-3lgvJ1?usp=sharing

Aniket Tendulkar	2020A7PS0001G
Rushabh Shah	2020A7PS1004G
Siddhant Sattyanaik	2020A7PS1009G
Pratham Bhatnagar	2020A7PS1222G

PROBLEM STATEMENT

The main goal of the IPARC challenge is to find whether there exists an inductive program learner (program) that can learn the human understandable function that maps the input image to an output image, given a set of pre-defined operations (Dilation and Erosion) and a set of 8 Structuring Elements and Color Bands. To this extent in Category A, the learner must be able to correctly predict the order of the structuring elements used given a Task. The first step would be to identify the position of the Structuring Element(SE) in the given input/output task.

HYPOTHESIS

We propose the following hypothesis: Given an input and output image mapping, the function(Dilation/Erosion) and the Structuring Element(SE), can we train a machine learning algorithm to find out the exact position of the SE in the series of the entire function? The algorithm should also be able to predict if the given SE was not used at all. If this is possible for every SE(to a good accuracy for every Task), we should be close to predicting the actual function that maps the input image to the output image given an IPARC challenge Task.

METHODOLOGY

Forming the Experiment

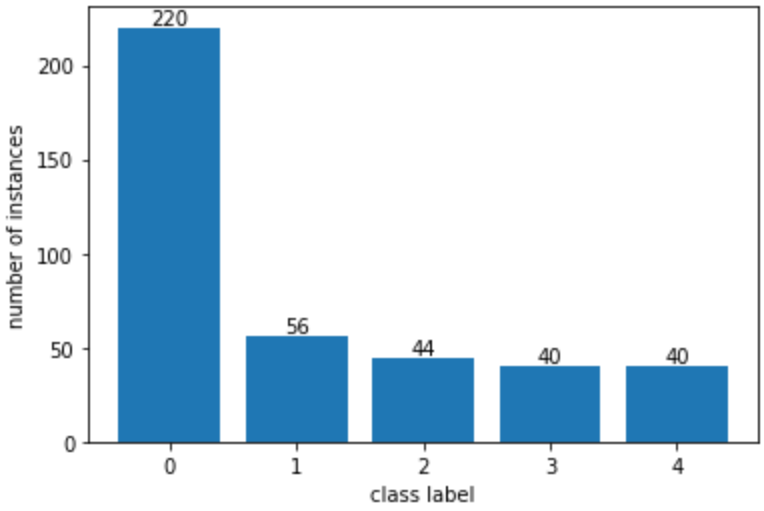
1. To correctly predict the location of the structuring element in the final function of the IPARC challenge Task, we take in user input for 'Dilation/Erosion' as well as 'SE_number"; the SE element number from 1 to 8 to check.
2. We test the machine learning algorithms used in this course, including Linear Models, Decision Trees and Support Vector Machines. We also use concepts including Cross-Validation, Bagging, Gradient Boosting and Feature Selection to check the hypothesis.
3. We check the accuracy of each of these models and then draw conclusions regarding our hypothesis with respect to the IPARC Challenge. For Category A(Easy), each solution has 4 Dilation functions followed by 4 Erosion Functions, each having a SE associated with it.

Data Processing

1. For each task, we have 4 input - output images, with a solution to the Task. Given the user input above we get the class labels of the dataset as follows: '0' representing SE not present, '1' representing the first index, '2' representing the 2nd position and so on for 4 positions for either Dilation or Erosion. So finally we have 5 class labels for the experiment.

2. For the features, each input and output image is a 15 X 15 image, so we have a total of 450 binary pixel features for the input to the learning algorithm (both input and output image). We also added the pixel features of the specific SE to the input features as well. We performed the experiments both with and without this 'background information'; and there was a large change in accuracy between the two. This 'background information' is vital to be fed to the learning algorithm. So we have a total of 459 features to the learning algorithm.
3. Each of the 100 tasks has 4 input-output pairs. So there are a total of 400 data points to the learning algorithm.

EXPERIMENTAL RESULTS AND VALIDATION



All the experimental results are for 'Dilation' as the function, 'SE6' as the Structuring Element. The distribution of the class labels is shown to the left. This represents an imbalanced dataset, with very less data compared to the feature space of the input of the data. So we have problems of imbalancing and very less data.

Algorithms and Techniques Used:

I. Principal Component Analysis(PCA):

To perform dimensionality reduction of the feature space, we use PCA on the input data. We first mean center the data and then plot the cumulative explainative variance with the number of components. We select 136 as the new dimension of the feature space where 95% of the variance among the feature space is retained, and then transform the data.

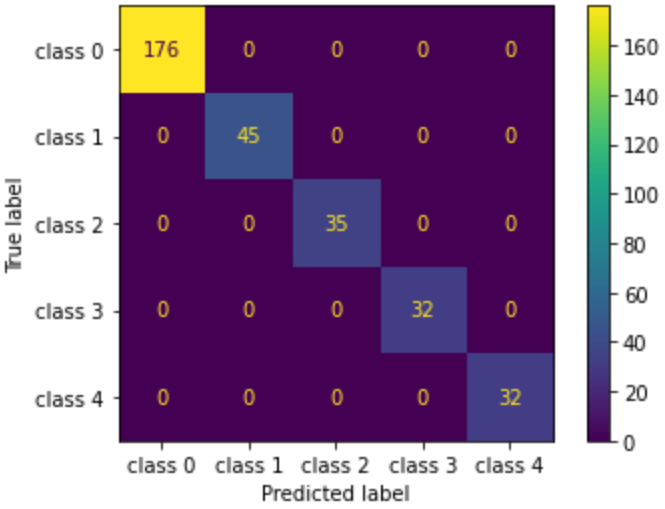
II. Train-Test Split:

We perform a 80 - 20 spilt on the input data and features after PCA transformation, also maintaining a random seed of 42 and shuffling the data. Also the split is 'stratified', to ensure that the test and

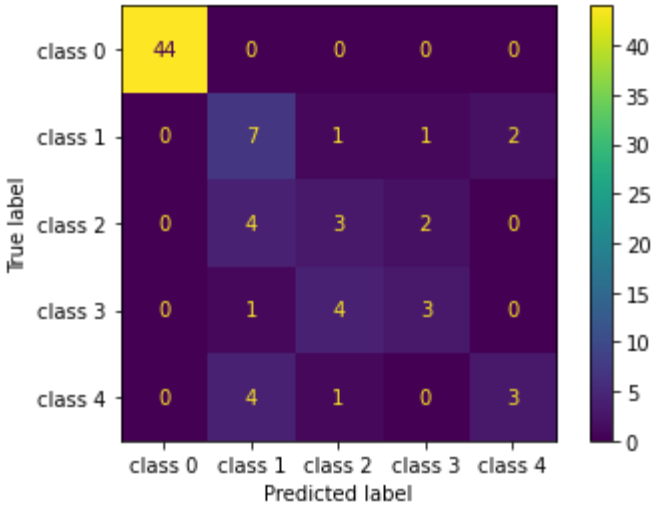
training data both have the same ratios of the class labels. So the training data is of shape (320, 136) and the test data is of shape (80, 136).

III. Logistic Regression

Each class was weighted inversely in proportion to its ratio in the training dataset. To use L2 regularization, we performed Cross Validation on the parameter 'C' for logistic regression. We performed 5- fold cross validation on the dataset, with L2 regularization, and selected the best model based on accuracy, which gave a 75% accuracy on the testing dataset. The model did overfit however, as visible in the Confusion matrix.



Train CM



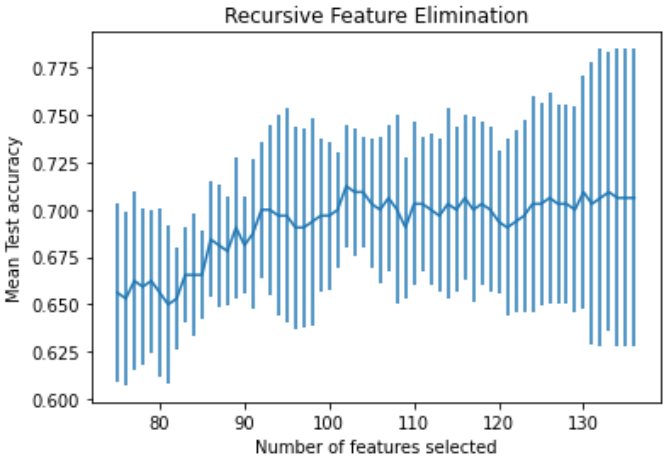
Test CM

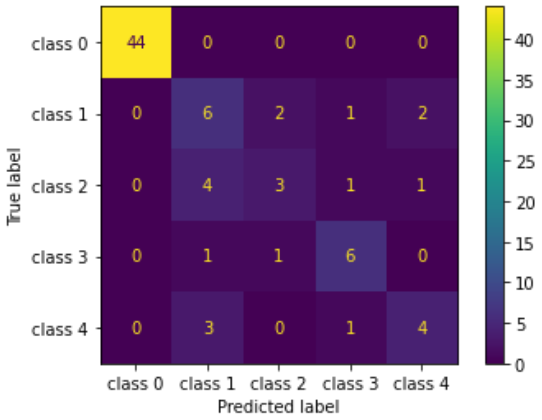
IV. Feature Selection (RFE and SFE)

For selecting the best subset of features, we tried using Recursive Feature Elimination on the best logistic regression model above. Starting with a minimum feature number as 75, we obtain the following plot (next page) for cross validation accuracy (folds = 5) and a 79% accuracy on the test dataset as well. The confusion matrix is shown as well. Greedy Backward Sequential Feature Selection did not give much of an improvement on the cross validation accuracy.

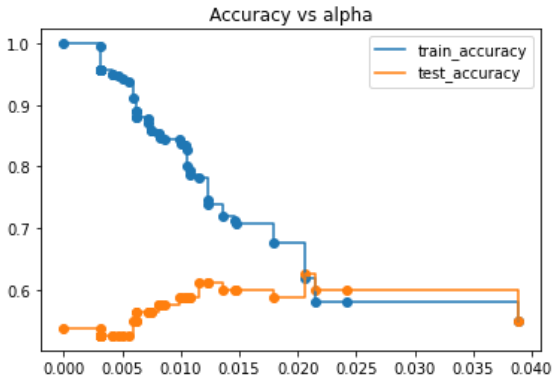
V. Decision Trees

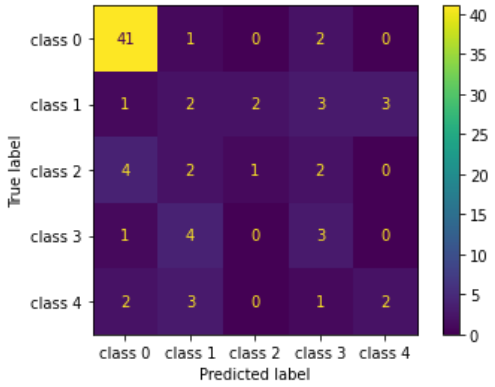
For the next hypothesis class, we tried fitting a single decision tree to the dataset. We allowed the tree to overfit, and then used pre pruning techniques (by cross validating parameters like max_height, min_samples_split, min_samples_leaves) and then using minimum cost pruning (by using the parameter ccp_alpha to prune the leaves of the





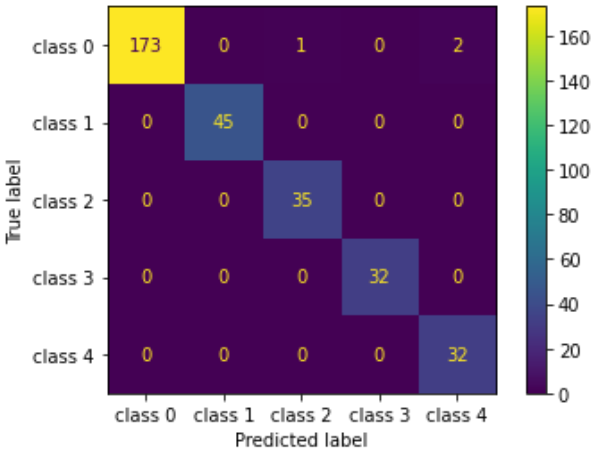
decision tree). We start by pruning leaves according to the impurity, till there is only 1 internal node. We obtain a test accuracy of 61.25% (up from 53.75% of the decision tree and then 55% after pre - pruning).



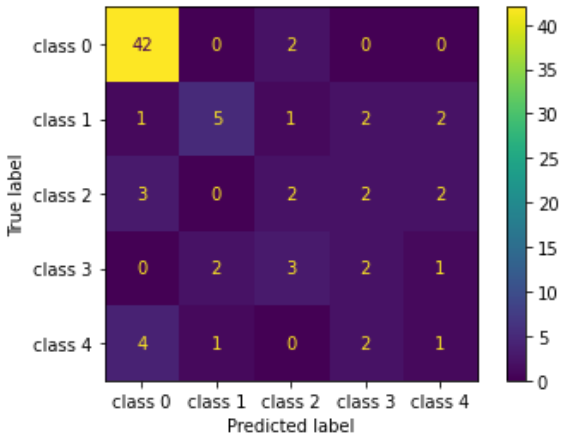


VI. Bagging and Boosting

To reduce the variance of the decision tree hypothesis, we used RandomForest and ExtraTrees Classifiers as bagging techniques. We performed both Randomized and GridSearch cross validation on the parameters : `n_estimators`, `max_features`, `max_depth`, `min_samples_split`, `min_samples_leaf`, `bootstrap`. For boosting, we used the XGBoost algorithm for gradient boosting using GridSearch for fine tuning the parameters as above. Both the methods yielded 65% accuracy on the test dataset. Interestingly, the models still overfit on the training dataset, despite the depth being set to 1 in the boosting algorithm. There was only a slight reduction in variance of the model and the confusion matrix shows that 'class 0' was being predicted more often.



Train CM



Test CM

VII. Support Vector Classifier(SVC)

Again we used the class weights to train the SVC with Cross Validation on the parameters kernel, tolerance level and L2 regularization parameter 'C'. Training with regularization and 3 fold Cross Validation, we obtain an accuracy of 75% on the test dataset. We also tried a pipeline with the Support Vector Classifier for feature selection and then using logistic regression for classification.

CONCLUSION

We observe that all of these models greatly overfit to the given data, due to the small size of the dataset. Decision trees overfit much more than Logistic Regression and SVC, even after trying different algorithms such as pruning, random forest and boosting with them. Due to the multi-class classification and the number of features, the complexity required from the models was too high to be properly trained on the small dataset. With our representation of data and algorithms chosen, we were unable to correctly predict the classification of the SE at a usable accuracy.