BITS F464 Machine Learning

Project Report – Group 12

Title - Relative learnability of operators sequence representing different control structures considered by the Böhm–Jacopini theorem

- Madhav Bajaj 2019B3A70256G f20190256@goa.bits-pilani.ac.in
- Ayush Vinod Sawant 2019B3A70615G f20190615@goa.bits-pilani.ac.in
- Shikhar Sahay 2019B3A70798G f20190798@goa.bits-pilani.ac.in
- Himanshu Daga 2019B3A70320G f20190320@goa.bits-pilani.ac.in

Github Link: https://github.com/Tricky308/ML_Project_G12

Problem Statement

Given an input image and output image pair from the IPARC Category B Hard dataset, where the input image can be transformed into the output image by a sequence of operators $\phi_1\phi_2 \ldots \phi_k\gamma$ of constant length k, where a sequence of operator(s) $\phi_1 \ldots \phi_i$ represents the Selection control structure, $\phi_{i+1} \ldots \phi_j$ represents the Iteration control structure, $\phi_{j+1} \ldots \phi_k$ represents the Sequence control structure, and γ represents the color change rule, i and j being constant; which sequence of operators among the three control structures can be predicted most accurately by the current ML models with a sufficiently large dataset?

Methodology

A. Data Preparation

1. The first step was to represent the target program through a dense data structure while retaining the information regarding the sequence of operators and the control structure they represented. The Category B Hard dataset is ideal for this experiment. Every solution to a Category B Hard question follows a definite pattern. The first operation is always a HitOrMiss operation representing the selection control structure. The band-1 from the HitOrMiss operation is then acted upon by a sequence of operations representing the iteration control structure. The band-2 is acted upon by a sequence of operations representing the sequence control structure. A constant color change rule is then applied at the end. Since the color change operator is constant, it is disconsidered and is not included in the target variable. Reviewing the *GenerateCatB_Hard.py* file, it is clear that the target problem can be represented by a 6-element array. Thus the following structure has been used to model the target variable Y. A custom json write function was used for used achieve the same.

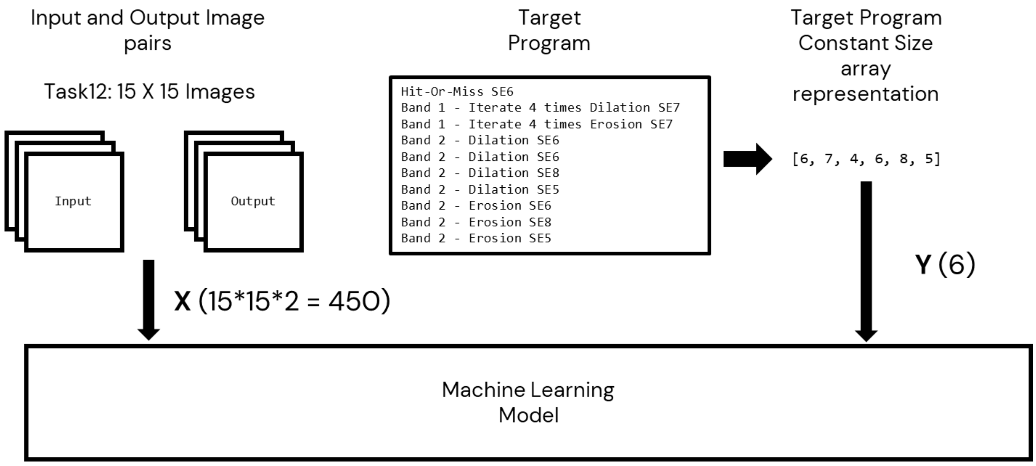
[HitOrMiss Operator SE number , Iterated operation SE number , Iterated operation k value , Sequence operator 1 SE number , Sequence operator 2 SE number , Sequence operator 3 SE number]

2. Since the objective here is to determine operators for which control structure can be learned effectively and not the few-shot learning aspect of the IPARC challenge, an enlarged dataset has been constructed for the machine learning models. Multiple Category B Hard datasets were

built with varying input-output image sizes (the target program structure remained the same) and the number of examples per task.

Dataset Number	Image Size	Number of Examples per Task
1	15 X 15	50
2	12 X 12	30
3	8 X 8	15
4	6 X 6	8
5	15 X 15	100
6	12 X 12	60
7	8 X 8	30
8	6 X 6	16

3. Input or the feature set X consisted of concatenated flattened input-output grids.



B. Model selection and Training

There are 8 structuring elements (SEs), thus there are 8 variations for all the operators in the target variable. The maximum value for the iteration number k is 4, thus 4 possible values. This problem can be seen as a multi-output multi-class (multi-task) classification problem. Five models out-of-the-box models from the sklearn library which support multi-task classification were considered:

1. Decision Tree Classifier
2. Random Forest Classifier
3. Extra Trees Classifier
4. KNeighbors Classifier
5. RadiusNeighbors Classifier

Hyperparameter optimization for the last four models was conducted on Dataset numbers 1 and 5 to get the best parameters for both sample size variations. Since sklearn does not support metrics calculation, and consequently, GridSearchCV or RandomizedSearchCV for multi-task classification, the optimization exercise was conducted manually. The model was trained with a train-test split of 75:25

Model (Hyperparameter)	Dataset Number 1	Dataset Number 2
Random Forest (number of trees)	250	250
Extra Trees (number of trees)	250	250
KNeighbors (number of neighbors)	3	3
RadiusNeighbors (radius)	25	25

Optimal Hyperparameter values

A handmade accuracy function to implement the metric was built, which returned the individual accuracy for each element of the target vector. Overall accuracy for the prediction of operators representing selection structure is given by the accuracy for the first element of the target vector. Similarly, for the iteration structure, is it given by the mean of accuracy of the second and third element. For sequence structure, it is given by the mean of the accuracy of the fourth, fifth, and sixth elements.

Experimental Results and Validation

Dataset Number	Decision Tree			Random Forest			Extra Trees		
	Selection	Iteration	Sequence	Selection	Iteration	Sequence	Selection	Iteration	Sequence
1	30.96%	57.44%	28.03%	61.28%	70.04%	41.71%	68.56%	71.44%	44.27%
2	33.87%	55.80%	29.07%	57.20%	73.33%	45.11%	64.93%	74.60%	48.53%
3	60.00%	56.67%	35.91%	75.73%	67.47%	47.11%	81.87%	68.40%	48.18%
4	81.50%	55.50%	51.50%	88.00%	59.25%	58.50%	91.00%	58.75%	59.83%
5	43.84%	60.16%	32.40%	67.64%	74.08%	48.12%	75.48%	75.90%	51.44%
6	42.80%	64.00%	35.31%	72.67%	80.03%	53.09%	79.87%	81.53%	56.33%
7	67.33%	67.13%	44.40%	81.87%	74.73%	55.16%	86.80%	76.40%	57.87%
8	85.00%	52.25%	57.33%	90.50%	51.50%	61.00%	92.00%	53.00%	60.58%
Average	**55.66%**	**58.62%**	**39.24%**	**74.36%**	**68.80%**	**51.22%**	**80.06%**	**70.00%**	**53.38%**
Overall Model Accuracy	**58.13%**			**73.13%**			**76.04%**		

Dataset Number	Kneighbors			RadiusNeighbors		
	Selection	Iteration	Sequence	Selection	Iteration	Sequence
1	44.56%	36.64%	22.99%	19.60%	29.28%	18.35%
2	54.40%	41.27%	25.91%	26.53%	26.53%	19.69%
3	75.20%	57.07%	41.16%	28.80%	31.20%	25.07%
4	86.50%	52.25%	55.50%	52.00%	25.25%	33.00%
5	51.96%	38.42%	24.57%	24.68%	29.44%	17.64%
6	62.00%	49.33%	29.44%	18.93%	29.87%	19.18%
7	77.60%	63.53%	45.82%	26.80%	27.60%	22.44%
8	90.75%	48.38%	55.58%	36.50%	29.13%	29.58%
Average	**67.87%**	**48.36%**	**37.62%**	**29.23%**	**28.54%**	**23.12%**
Overall Accuracy	**55.49%**			**31.13%**		

	Best Accuracy across all models	Average Accuracy across all models	Best of best elemental accuracy within the sequence of operators across all models	Average of best elemental accuracy within the sequence of operators across all models
Selection	80.06%	61.44%	-	-
Iteration	70.00%	54.86%	83.90%	69.00%
Sequence	53.38%	40.92%	61.52%	48.73%

Final Results

- From the results, it can be inferred that with a structured target program representation and enough data, the current machine learning models can achieve a reasonable level of accuracy on unseen tasks suggesting a modest degree of generalizability. Overall, the Extra Trees model achieved the highest accuracy.

- While in general, the accuracy for the operators sequence for control structure selection and sequence increased as image size and thus the feature matrix to the model was reduced but this was not the case for the operator sequence for control structure iteration where accuracy decreased. Accuracy increased for all models as the number of samples per task increased.
- Prediction accuracy for the selection control structure was superior compared to iteration and sequence in terms of both average and highest accuracy observed. Similarly, better results were obtained for iteration compared to the sequence control structure.
- While the above result could be attributed to the fact that the iteration measure is the average accuracy of two elements in the target vector and the sequence measure is the average accuracy of three elements in the target vector, when a similar analysis is conducted for the max of two and three elements for iteration and sequence measure respectively, the results obtained for the iteration control structure were superior to the selection and sequence.

Conclusion and Future Work

It can be concluded from the results that under the conditions stated for this experiment, current ML models can predict the operator sequence for selection and iteration control structure explicitly better than for sequence control structure, but no comment could be made regarding the ordering between themselves.

This experiment could important results for program synthesis through Inductive Logic programming engine (ILP) or other means, specifically for the trade off decisions between increasing the size of background relations with the addition of new functions and the decrease in overall size of the programs for all tasks (For all tasks in this category, systems are given a set of tasks $\{T_1, T_2, \dots, T_N\}$. Let $H_1, H_2, \dots, H_n$ be individual programs obtained independently for each task using background relations in Σ_1. The tasks involve automatically augmenting Σ_1 to $\Sigma_1^+ \supset \Sigma_1$ and obtaining programs $H'_1, H'_2, \dots, H'_n$ such that

$$\left|\Sigma_1 \bigcup_{1}^{N} H_i\right| > \left|\Sigma_1^+ \bigcup_{1}^{N} H'_i\right|$$

This experiment could help ILP engine to choose between different predicate based on different control structures to add to the background relations. Experiments could be devised to confirm the suggested hypothesis.