Experiment Section

June 18, 2020

1 Experimental Evaluation

After adding the multi-phase ranking function learning algorithm to SVMRanker, we conducted experiments on constructed dataset of loop programs to compare the algorithms of learning nested ranking function, of multi-phase ranking function and algorithm in the state-of-the-art tool LassoRanker embedded in Ultimate Automizer. Our dataset are 230 Boogie programs, where 96 programs are converted from library of sv-comp**??** and 134 loop programs in Boogie are grabbed from repository of Ultimate Automizer. For the configuration of experments, we use a server with a 3.6 GHz Intel Core i7-4790 CPU and 16GB RAM and timeout is set to 300 seconds for each case.

1.1 Overview of the Experiments

As shown in Table 1, we use our 230 examples as inputs to SVMRanker and LassoRanker. For SVMRanker, we use "Nested, 2-Multi, 4-Multi" to represent the sythesising of nested ranking function, 2-phases-bounded and 4-phases-bounded multiphase ranking function respectively. lx: Add description of the bound if the phase bound is not mentioned previously :xl In the experiment, the algorithm tries different number of phases of nested ranking function according to the result of the

	SVMRanker			LassoRanker
	Nested	2-Multi	4-Multi	
Terminating	54	72	78	72
Non-teminating	50	50	50	52
Unknown	126	108	102	106
Total Time	196s	1183s	4693s	2053s

Table 1: General Experiment Results

	Nested	2-Multi	4-Multi
Sampling time	14.9s	59.6s	170.3s
Training time	7.2s	65.1s	186.9s
Z3 Solving time	24.9s	278.6s	1965.7s

Table 2: Detailed Time Cost of SVMRanker

	Non-linear loop programs
SVMRanker	13
LassoRanker	0
Total number	19

Table 3: Solved Number of Non-linear Loops

learning. Furthermore, we use linear and non-linear templates for nested ranking function and use only linear templates for multiphase learning.

From Table 1, it is obvious our new algorithm for multiphase r.f. is more powerful and can solve more cases than that of nested r.f. learning. From the comparation between "2-Multi" and "4-Multi", it is clear that larger bound on phases is given, more ranking functions can be found. Besides, since the number of terminating cases of multiphase learn is almost the same as LassoRanker, we can tell that our multiphsae learning algorithm enhance the capability of SVMRanker to deal with linear loops programs in our data set.

As for the total running time of this experiment. The total time of multiphase r.f. learning is much more than nested r.f. learning. This can be attribute to the backtracking and incremental learning of multiphase.lx: Add description of backtracking and incremental learning if the phase bound is not mentioned previously :xl Thought the long running time, we are still optimistic about our tool that it solves the same number of terminating cases as LassoRanker but only uses about half of their time.

1.2 Detailed Evaluation

1.2.1 *(a) Time cost of each stage.*

To better illustrate the time cost, we record sampling time, training time and Z3 sovling time of the experiments as shown in Table 2. Time costs all increase fastly when the depth bound of multiphase learning increases. Most of the time are used on solving Z3 formulae to check the satisfiability of requirements on learned functions. In the future, since our multiphase templates are all linear templates, we wish to improve the result by using more efficient LP tools like CPLEX and PuLP??.

1.2.2 *(b) Advantage on non-linear loops.*

According to Table 3, we summarized the non-linear cases solved by our tool and LassoRanker. We state that our tool performs better than LassoRanker on non-linear loops. This is because the latter only utilize linear templates while our algorithm use non-linear templates as well.

lx: exp section is slightly long, to cut:xl