PURLTL: Mining LTL Specification from Imperfect Traces in Testing

Bo Peng[1] Pingjia Liang[1] Tingchen Han[1] Weilin Luo[1]
Jianfeng Du[2,3] Hai Wan[1] Rongzhen Ye[1] Yuhang Zheng[1]

[1]Sun Yat-Sen University

[2]Guangdong University of Foreign Studies

[3]Pazhou Lab

The 38th IEEE/ACM International Conference on Automated Software Engineering, September 2023

Problem

Linear Temporal Logic (LTL) specification mining is to mine LTL formula which describes the system behaviors from traces.

Input

A set of traces:

$$[a, a, b, c],$$
$$[b, a],$$
$$\ldots.$$

Output

An LTL formula: $a\mathrm{U}b$.

Challenges

Large search space

- Exponential complexities
- Requiring templates

Noisy data

- Lots of traditional approaches failed with noisy traces
- Very common in industrial scenarios, may come from
  - buggy programs
  - partial profiling
  - ...

Neural Network as LTL Path Checking

LTL path checking

- Input: a single trace
- Output: satisfy/unsatisfy a certain LTL formula

Examples

Formula: $a \mathrm{U} b$

- Input 1: $[a, a, b, c]$
- Output 1: SAT

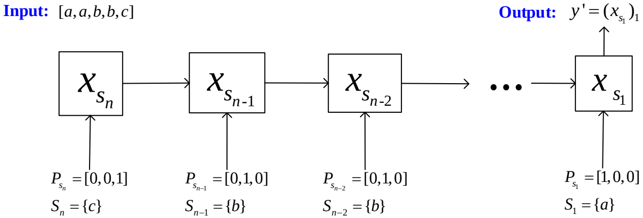
- Input 2: $[a, b, a, c]$
- Output 2: UNSAT

Neural Network as LTL Path Checking

We design a neural network to simulate the LTL path checking:

$$f_\theta(x) : T \to \{0, 1\}$$

where θ is a set of trainable parameters, and T is the trace space.



After training, we can interpret θ to an LTL formula.

Training Strategy

New chellenge

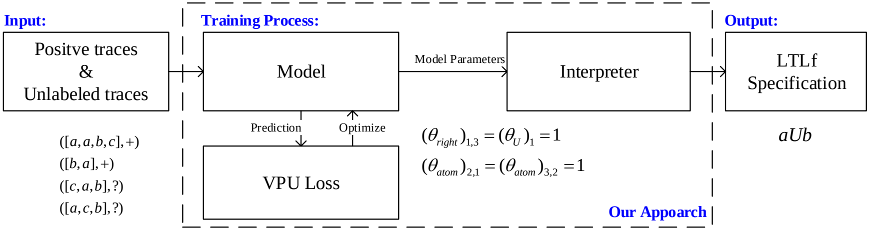
- We need a label-balanced training set for binary classification tasks
- In this case, negative samples are unavailable

Solution

Apply positive and unlabeled (PU) learning to our approach.

- Use real software logs as positive sample
- Randomly generated traces as unlabeled samples
- Apply variational PU loss

The Overall Procedure of Our Approach



Results on Noise-free Inputs

Table: F_1 scores (%) of `Texada`, `GLTLf` and `PURLTL` on noise-free data.

	Texada	GLTLf	PURLTL
Response	100	49.41	100
Alternating	100	81.97	89.49
MultiEffect	100	80	99.5
MultiCause	100	86.58	100
EffectFirst	100	58.61	99.5
CauseFirst	100	85.11	100
OneCause	100	66.52	100
OneEffect	100	61.98	100

Results on Noisy Inputs

Table: F_1 scores (%) of GLTLf and PURLTL with different noise rates.

		0%	10%	30%	50%
Response	GLTLf	49.41	49.25	49.25	49.25
	PURLTL	<u>100</u>	<u>100</u>	<u>100</u>	<u>100</u>
Alternating	GLTLf	81.97	81.97	80	80
	PURLTL	<u>89.49</u>	<u>89.49</u>	<u>89.49</u>	<u>89.49</u>
MultiEffect	GLTLf	80	80	80	80
	PURLTL	<u>99.5</u>	<u>99.25</u>	<u>97.3</u>	<u>97.3</u>
MultiCause	GLTLf	86.58	86.58	86.58	86.58
	PURLTL	<u>100</u>	<u>100</u>	<u>100</u>	<u>100</u>
EffectFirst	GLTLf	58.61	58.61	58.61	58.61
	PURLTL	<u>99.5</u>	<u>99.5</u>	<u>96.37</u>	<u>96.37</u>
CauseFirst	GLTLf	85.11	85.11	85.11	85.11
	PURLTL	<u>100</u>	<u>100</u>	<u>100</u>	<u>100</u>
OneCause	GLTLf	66.52	66.52	66.52	66.52
	PURLTL	<u>100</u>	<u>100</u>	<u>100</u>	<u>100</u>
OneEffect	GLTLf	61.98	61.98	61.98	61.98
	PURLTL	<u>100</u>	<u>92.35</u>	<u>92.35</u>	<u>95.69</u>

Ablation Study of VPU Loss

Table: F_1 scores (%) of `PURLTL` with different loss functions.

	Response	Alternating	MultiEffect	MultiCause
MSE	49.41	88.89	88.79	86.58
VPU	66.42	98.04	96.11	86.58
	EffectFirst	**CauseFirst**	**OneCause**	**OneEffect**
MSE	75.14	85.11	61.3	72.49
VPU	61.21	85.11	83.68	72.49

Results with and without Templates

Table: F_1 scores (%) on noise-free inputs with and without templates.

	Response	Alternating	MultiEffect	MultiCause
w/ t.	100	100	99.5	100
w/o t.	66.42	98.04	96.11	86.58
	EffectFirst	**CauseFirst**	**OneCause**	**OneEffect**
w/ t.	99.5	100	100	100
w/o t.	61.21	85.11	83.68	72.49

Conclusions

We propose an approach to mine LTL specifications, namely PURLTL, which has the following advantages:

- High efficiency (polynomial complexity of neural networks)
- Ability of accepting noisy inputs
- Template-free