



Highlights 2022

of Logic, Games and Automata

Paris, June 28–July 1st, 2022

Scalable Anytime Algorithms for Learning Fragments of Linear Temporal Logic (SCARLET)

Ritam Raha Rajarshi Roy Nathanaël Fijalkow Daniel Neider



Universiteit
Antwerpen

LaBRI



MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Appeared at TACAS'22

Explainable AI

Explainable AI

Goal: Learn simple (human interpretable) models by observing complex systems

Robot Motion-Planning

Robot Motion-Planning

Positive

Negative

Robot Motion-Planning

Positive

Negative

A ^ Finally B

LTL as a descriptive model

Linear Temporal Logic

Eg. Globally, Finally, Next

Syntax:

$$\varphi ::= p \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid X\varphi \mid F\varphi \mid G\varphi \mid \varphi_1 U \varphi_2$$

LTL as a descriptive model

Linear Temporal Logic on finite words (Vardi & Giacomo '13)

Eg. Globally, Finally, Next

Syntax:

$$\varphi ::= p \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid X\varphi \mid F\varphi \mid G\varphi \mid \varphi_1 U \varphi_2$$

LTL as a descriptive model

Linear Temporal Logic on finite words (Vardi & Giacomo '13)

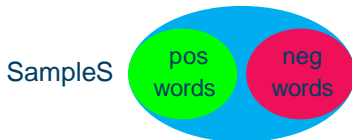
Eg. Globally, Finally, Next

Syntax:

$$\varphi ::= p \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid X\varphi \mid F\varphi \mid G\varphi \mid \varphi_1 U \varphi_2$$

The fragment: LTL(F, X, G, ^, _)

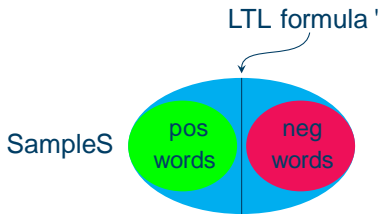
The learning problem



LTL Learning on Finite Words

Input: A set of positive words P & negative words N

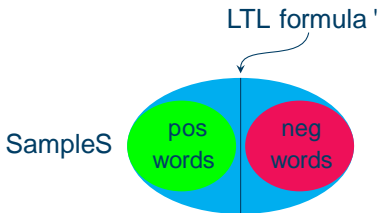
The learning problem



LTL Learning on Finite Words

Input: A set of positive words P & negative words N

The learning problem



LTL Learning on Finite Words

Input: A set of positive words P & negative words N

Question: Find a minimal LTL formula ϕ such that,
 $\forall w \in P; w \models \phi$ and $\forall w \in N; w \not\models \phi$?

Theorem (Fijalkow & Lagarde '21)

The learning problem for the fragments of LTL: $LTL(X)$, $LTL(F, \wedge)$ and $LTL(F, X, \wedge, _)$ is NP-complete.

Theorem (Fijalkow & Lagarde '21)

The learning problem for the fragments of LTL: $LTL(X)$, $LTL(F, \wedge)$ and $LTL(F, X, \wedge, _)$ is NP-complete.

Existing approaches:

- | SAT-Solvers -FLIE (Neider & Gavran '18)
- | SyGuS solvers SYSLITE(Arif et al. '20)

Theorem (Fijalkow & Lagarde '21)

The learning problem for the fragments of LTL: $LTL(X, \wedge)$, $LTL(F, \wedge)$ and $LTL(F, X, \wedge, _)$ is NP-complete.

Existing approaches:

- | SAT-Solvers -FLIE (Neider & Gavran '18)
- | SyGuS solvers SYSLITE(Arif et al. '20)

Overview

- | For all LTL formulas of size k , check if separating.
- | Increase k and repeat.

Towards Approximation

Towards Approximation

Overview.

- I Extract LTL patterns of increasing complexity from sample
(Technique used: Dynamic Programming)

Towards Approximation

Overview.

- | Extract LTL patterns of increasing complexity from sample
(Technique used: Dynamic Programming)
- | Generate their Boolean combinations to find the (minimal) formula
by solving Boolean Set Cover problem
(Technique used: Greedy approximation or Decision Tree)

Finding LTL patterns

Sample S

Positive Words

pqqp

qqpp

Negative Words

qqqq

ppqp

Idea:

Candidate:

Formula:

Finding LTL patterns

Sample S

Positive Words

pqqp

qqpp

Negative Words

qqqq

ppqp

Idea: Find separating patterns with intervals

Candidate:

Formula:

Finding LTL patterns

Sample S

Positive Words

pqqp X

qqpp X

Negative Words

qqqq

ppqp

Idea: Find separating patterns with intervals

Candidate: $(1, q, > 0, p)$

Formula:

Finding LTL patterns

Sample S

Positive Words

pqqp X

qqpp X

Negative Words

qqqq 8

ppqp 8

Idea: Find separating patterns with intervals

Candidate: $(1, q, > 0, p)$

Formula:

Finding LTL patterns

Sample S

Positive Words

pqqp X

qqpp X

Negative Words

qqqq 8

ppqp 8

Idea: Find separating patterns with intervals

Candidate: $(1, q, > 0, p)$

Formula: $X(q \wedge Fp)$

Directed LTL

LTL patterns that arise from the following grammar:

$$\phi := X^n p \mid FX^n p \mid X^n(\phi \wedge \phi') \mid FX^n(\phi \wedge \phi');$$

Theorem

The boolean combination of dLTL formulas is as expressive as LTL($F, X, \wedge, _$)

Directed LTL

LTL patterns that arise from the following grammar:

$$\phi ::= X^n p \mid \neg \phi \mid \phi X^n \phi \mid \phi X^n (\phi \wedge \psi) \mid \phi X^n (\phi \vee \psi);$$

Theorem

The boolean combination of dLTL formulas is as expressive as LTL(F, X, \wedge, \vee)

Dual: $\phi : F : \psi = G : \psi : \phi$: swap positive and negative words!

Directed LTL

LTL patterns that arise from the following grammar:

$$\phi ::= X^n p \mid \neg \phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \wedge \psi \mid \phi \vee \psi;$$

Theorem

The boolean combination of dLTL formulas is as expressive as LTL(F, X, \wedge, \vee)

Dual: $\neg \phi : F : \neg \phi = G : \neg \phi$: swap positive and negative words!

Boolean Set Cover

Boolean Set Cover

Problem: Find the minimal boolean combination of formulas that separates the sample

Boolean Set Cover

Problem: Find the minimal boolean combination of formulas that separates the sample

Sol: $(f_1 \wedge f_2) \vee f_3$

Boolean Set Cover

- I A similar greedy approximation algorithm to classical set cover

Boolean Set Cover

- | A similar greedy approximation algorithm to classical set cover

- | Another approach: Decision Trees

Advantages of our approach

- | Anytime algorithm

Advantages of our approach

- | Anytime algorithm
- | Optimized according to the Sample

Advantages of our approach

- | Anytime algorithm
- | Optimized according to the Sample
- | Noisy Data Setting

SCARLET

Future Work/ Open Questions

Future Work/ Open Questions

- | Exact approximation factor of the algorithm

Future Work/ Open Questions

- | Exact approximation factor of the algorithm
- | Capture more expressive power: learn formulas with U-operator

Future Work/ Open Questions

- | Exact approximation factor of the algorithm
- | Capture more expressive power: learn formulas with U-operator
- | Towards real-valued traces: learn formulas in STL

