Proving hardness of LTL Learning

Joint work with
Nathanaël Fijalkow
Guillaume Lagarde

GT DAAL 2024

LTL on finite words (LTL_f)

$$\varphi ::= a \mid \neg a \quad \} \text{ atoms}$$

$$\mid \varphi \vee \varphi \mid \varphi \wedge \varphi \quad \} \text{ boolean operators}$$

$$\mid X\varphi \mid F\varphi \mid G\varphi \mid \varphi U \varphi \mid \varphi R \varphi \quad \} \text{ temporal operators}$$

$a \longrightarrow a \ \_\ \_\ \_\ \cdots$

$X\varphi \longrightarrow \_\ \varphi\ \_\ \_\ \cdots$

$F\varphi \longrightarrow \_\ \cdots\ \varphi\ \_\ \_\ \_$

$G\varphi \longrightarrow \varphi\ \varphi\ \varphi\ \varphi\ \varphi \cdots \varphi$

$\varphi \cup \psi \longrightarrow \varphi \cdots \varphi\ \psi\ \_\ \_\ \_\ \cdots$

$abab \models G(b \vee Xb)$

$aab \models GFb$

$aab \models Xa \cup Xb$

$LTL_f[O_p]$ learning problem

babaaba
bbaabba

Positive words

bababba

Negative words

Is there a formula of size $\leq k$ that
- accepts all positive words
- rejects all negative words

?

Example

babaaba
bbaabba

$+$

bababba

$-$

$Xb \lor X^4 a \rightarrow 8$

$X(b \lor X^3 a) \rightarrow 7$

$F(a \land Xa) \rightarrow 5$

Minimal automaton learning problem

Input: $u_1, \dots, u_n, v_1, \dots, v_m, k$

Output: $\exists \mathcal{A}$ DFA of size k that $\begin{cases} \text{accepts all } u_i \\ \text{rejects all } v_j \end{cases}$

Gold '78 This problem is NP-complete.

Pitt & Warmuth '93 Unless P = NP, there is no polynomial-time algorithm computing a separating automaton of size poly(opt) with poly a polynomial.

$LTL[Op]$ is NP-complete for all* sets of operators Op.

*except trivial ones

Hitting Set Problem

Input: sets $S_1, \dots, S_n \subseteq U$, $k \in \mathbb{N}$

Output: $\exists ?\ H \subseteq U$, $|H| \leq k$,

$\forall i,\ H \cap S_i \neq \emptyset$

H

$U = \{1\ 2\ 3\ 4\ 5\ 6\}$

	1	2	3	4	5	6
S_1	0	1	0	0	1	0
S_2	1	0	0	1	0	1
S_3	0	0	1	0	1	0
S_4	1	0	0	0	0	1

$$
\begin{array}{c|cccccc}
 & & & H & & & \\
U = \{ & 1 & 2 & 3 & 4 & 5 & 6 \} \\
\hline
S_1 & 0 & 1 & 0 & 0 & 1 & 0 \\
S_2 & 1 & 0 & 0 & 1 & 0 & 1 \\
S_3 & 0 & 0 & 1 & 0 & 1 & 0 \\
S_4 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}
\longrightarrow
\begin{array}{c|cccccc}
U = & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\
\hline
V_1 & a_1 & b_2 & a_3 & a_4 & b_5 & a_6 \\
V_2 & b_1 & a_2 & a_3 & b_4 & a_5 & b_6 \\
V_3 & a_1 & a_2 & b_3 & a_4 & b_5 & a_6 \\
V_4 & b_1 & a_2 & a_3 & a_4 & a_5 & b_6
\end{array}
\begin{array}{l}) + \\ \\ \Big) - \end{array}
$$

Best formula:

$$G\left(\bigwedge_{i \in H} \neg a_i\right)$$

For the $LTL[X, \wedge]$ fragment:

	a_1	a_2	a_3	a_4	a_5	a_6	
$u=$	a_1	a_2	a_3	a_4	a_5	a_6	+
v_1	a_1	b_2	a_3	a_4	b_5	a_6	−
v_2	b_1	a_2	a_3	b_4	a_5	b_6	−
v_3	a_1	a_2	b_3	a_4	b_5	a_6	−
v_4	b_1	a_2	a_3	a_4	a_5	b_6	−

Best formula

$$a_1 \wedge X^4 a_5$$ ~ Hitting Set

→ NP-complete

Formulas can be put in normal form

$$Xa \wedge X^3 b \wedge X^4 a \equiv X(a \wedge X^2(b \wedge Xa)) \equiv \overbrace{\_\ a\ \_\ b\ a\ \_\_\_}^{\text{pattern}}$$

Petr '96 There is a $O(\log(n))$-approximation for Hitting set...

Dinur & Steurer '14

... and no $O(\log(n)(1-o(1)))$ approximation.

Fijalkow & Lagarde '21 There is a $O(\log(n))$ approximation for $LTL[X,\wedge]$ learning but no $O(\log(n)(1-o(1)))$ approximation.

What about a fixed alphabet?

Fijalkow, Lagarde, M. '24 (ongoing)

$LTL_f[Op]$ is NP-complete for all* sets of operators

even over a fixed alphabet.

$Op \subseteq \{F, G, \land, \lor, X\}$

But we need new proof techniques ...

For some fragments, nothing to do

$u=$	a	a	a	a	a	a	+
v_1	a	b	a	a	b	a	−
v_2	b	a	a	b	a	b	−
v_3	a	a	b	a	b	a	−
v_4	b	a	a	a	a	b	−

Best formula

$$a \wedge X^4 a$$

↪ We extend it to show NP-hardness for all $\{X, \wedge\} \subseteq Op \subseteq \{F, G, X, \wedge, \vee\}$

Unary Stuttering fragment $LTL[F, G, \wedge, \vee]$

→ Hard to count:

To distinguish $c\,ab\,ab\,ab \ldots ab\,ab\,c$

and $c\,\underbrace{ab\,ab\,ab \ldots ab}_{M \text{ times}}\,c$

We need $\underbrace{F(a \wedge F(b \wedge \ldots \wedge Fa))}_{\text{Size } 6M}$

Hitting Set Reduction

$U =$	$(ab)^{n+1}\, c\, (ab)^{n+1}\, c \cdots c\, (ab)^{n+1}$
V_1	$(ab)^{n}\, c\, (ab)^{n+1}\, c \cdots c\, (ab)^{n}$
V_2	$(ab)^{n+1}\, c\, (ab)^{n}\, c \cdots c\, (ab)^{n+1}$
V_3	$(ab)^{n+1}\, c\, (ab)^{n+1}\, c \cdots c\, (ab)^{n}$
V_4	$(ab)^{n}\, c\, (ab)^{n}\, c \cdots c\, (ab)^{n+1}$

$+$

$-$

Now prove it!

Goal → Prove that u and $v_1, \ldots, v_n$ are not separable by a formula of size $\leq k$

Ehrenfeucht-Fraïssé-like games ?

↳ [Etessami, Wilke '00] EF games for LTL

↳ well-suited for depth, not size

↳ Boolean operators are hard to handle

We only need (Duplicator wins $\Rightarrow$ no formula)

↳ Let Spoiler cheat!

Configuration $= \begin{cases} (x_i, y_i)_{i \in I}, & I \subseteq [\![1, n]\!] \\ p \in \mathbb{N} \end{cases}$

- At the start, $(u, v_i)_{i \in [\![1, n]\!]}$, k.
- Spoiler implicitly constructs a formula.

Spoiler can

→ play 'F' → S picks a suffix of each X_i

D " " " Y_i

$p \leftarrow p-1$

→ play 'G' → similar

→ play 'boolean' → S picks p_1, p_2 s.t. $p = p_1 + p_2 + 1$

S picks I_1, I_2 s.t. $I = I_1 \cup I_2$

D picks I_1, p_1 or I_2, p_2

Duplicator wins

↳ copycat strategy

If S goes too fast

c a b a b a b a b c

c a b a b a b c

D can catch up

Full LTL_f

Idea: $u = b\,a^{i_1}\,b\,a^{i_2}\,b\,a^{i_3}\,b\,a^{i_4}\,b$

$S_j = \{2, 4\} \rightarrow v_j = b\,a^{i_1}\,b\,a^{i_2+1}\,b\,a^{i_3}\,b\,a^{i_4+1}\,b$

Best formula: $F(b \wedge X^{i_2+1}(b \vee X^{i_4 - i_1 + 1} b))$

if $H = \{2, 4\}$

Future work:

- "Efficient" formula enumeration.
- Approximation algorithms for otha fragmets.
- Game characterisation ?

Thank you for your attention !