Cause Mining with STL

Ebru Aydin Gol
Computer Engineering, METU

139th Shonan Meeting
Power train control benchmark model by Toyota

Can we describe the cause of the undesired event?
Our Approach in a Nutshell

\[
\phi = \phi_1 \lor \phi_2 \lor \phi_3 \\
\phi_1 = P_{[1,1]}((x^1 > 15) \land (s^1 = 1) \land (s^0 = 0)) \\
\phi_2 = P_{[1,1]}((x^1 > 25) \land (s^1 = 1)) \\
\phi_3 = P_{[1,1]}((x^4 < 10) \land (s^1 = 1) \land (s^0 = 0))
\]
Outline

• Temporal logics
• Problem Definition
• Greedy Search for Formula Synthesis
• Iterative Search for Formula Synthesis
• Search for Controllable Formula
• Discussion
Linear Temporal Logic

Eventually $a$

$$F \; a$$

Next $a$

$$X \; a$$

Globally $a$

$$G \; a$$

$a$ Until $b$

$$a \; U \; b$$

$$\phi = T \mid F \; \phi \mid G \; \phi \mid \phi U \; \phi \mid X \; \phi \mid \phi \land \phi \mid \phi \lor \phi \mid \neg \phi$$

Boolean operators: $\land$, $\lor$, $\neg$

Temporal operators: $F$, $G$, $U$, $X$
Linear Temporal Logic

Eventually $a$

$F \ a$

Next $a$

$X \ a$

Globally $a$

$G \ a$

$a$ Until $b$

$a \ U \ b$

Boolean operators: $\land, \lor, \neg$

Temporal operators: $F, G, U, X$

Metric Temporal Logic

Real time, boolean predicates

$G_{[0,t]}(r \rightarrow F_{[0,c]}g)$
Linear Temporal Logic

Eventually $a$

$$F a$$

Next $a$

$$X a$$

Globally $a$

$$G a$$

Until $a$ $b$

$$a U b$$

Boolean operators: $\land$, $\lor$, $\neg$

Temporal operators: $F$, $G$, $U$, $X$

Metric Temporal Logic

$$F[a_{2,5}] q$$

$$G[a_{2,5}] a$$

$$a U[a_{2,5}] b$$

$$\phi = T | F \phi | G \phi | \phi U \phi | X \phi | \phi \land \phi | \phi \lor \phi | \neg \phi$$

Real time, boolean predicates

$$G_{[0,t]} (r \rightarrow F_{[0,c]} g)$$

Signal Temporal Logic

Real time, real valued predicates

$$G_{[0,t]} (r[t] > 0 \rightarrow F_{[0,d]} g[t] > 0)$$
Signal Temporal Logic

- Predicates are over real values (signals), real time
- Allows for quantitative semantics

$$\phi = T \mid \phi_1 U_{[a,b]} \phi_2 \mid \phi_1 \land \phi_2 \mid \neg \phi \mid x < c$$

$$F_{[a,b]} \phi = T U_{[a,b]} \phi$$

$$G_{[a,b]} \phi = \neg (F_{[a,b]} \neg \phi)$$
Past Time Signal Temporal Logic

\[ \phi = T \left( \phi_1 S_{[a,b]} \phi_2 \right) \phi_1 \land \phi_2 \mid \neg \phi \mid x < c \]

\[ P_{[a,b]} \phi = T S_{[a,b]} \phi \]

\[ A_{[a,b]} \phi = \neg (P_{[a,b]} \neg \phi) \]
Problem Formulation

Given a set of labeled signals, find a ptSTL formula describing the labeled events.

\[ x_{i}[t + 1] = x_{i}[t] + d_{i}[t] - f_{i}^{\text{out}}(x_{i}[t]) + \sum_{k \in \mathcal{L}_{i}^{\text{out}}} \beta_{k}f_{k}^{\text{out}}(x_{k}[t]) \]

1 - Simulate
2 - Label the traces
3 - Find the reason

Given a system and a function over the signals for labeling unwanted events, find the cause of these events.
Given a set of labeled signals, find a ptSTL formula describing the labeled events

\[ D = \{(T, L) \mid T = (t_0, x_0), \ldots, (t_n, x_n), L = (t_0, l_0), \ldots, (t_n, l_n)\} \]

\[ (L, t) = 1 \quad \quad (T, t) \models \phi \]
Literature on Formula Synthesis


“…find formulas defining signals as tight as possible…”

Zhaodan Kong, Austin Jones, Ana Medina Ayala, Ebru Aydin Gol and Calin Belta, Temporal Logic Inference for Classification and Prediction from Data, Hybrid Systems: Computation and Control (HSCC), Berlin, Germany, 2014

“…formula that can discriminate between the desirable behaviors and the undesirable ones is constructed…”


“…to infer signal temporal logic (STL) formulas that characterize the behavior of a dynamical system using only observed signal traces of the system …”

Giuseppe Bombara, Cristian Vasile, Francisco Penedo, Hirotoshi Yasuoka, Calin Belta, A Decision Tree Approach to Data Classification using Signal Temporal Logic, Hybrid Systems: Computation and Control (HSCC), Vienna, Austria, 2016


Problem Formulation

Given a set of labeled signals, find a ptSTL formula describing the labeled events

\[ D = \{(T, L) \mid T = (t_0, x_0), \ldots, (t_n, x_n), L = (t_0, l_0), \ldots, (t_n, l_n)\} \]

\( (L, t) = 1 \)
\( (T, t) \models \phi \)

\( (L, t) = 1 \)
\( (T, t) \models \neg \phi \)

\[ TP(D, \phi) = \sum_{(T,L) \in D} \sum_{t=0}^{n} \begin{cases} 1 & \text{if } l_i = 1 \text{ and } (T, t) \models \phi \\ 0 & \text{otherwise} \end{cases} \]

\[ FN(D, \phi) = \sum_{(T,L) \in D} \sum_{t=0}^{n} \begin{cases} 1 & \text{if } l_i = 1 \text{ and } (T, t) \models \neg \phi \\ 0 & \text{otherwise} \end{cases} \]

\[ TN(D, \phi) = \sum_{(T,L) \in D} \sum_{t=0}^{n} \begin{cases} 1 & \text{if } l_i = 0 \text{ and } (T, t) \not\models \phi \\ 0 & \text{otherwise} \end{cases} \]

\[ FP(D, \phi) = \sum_{(T,L) \in D} \sum_{t=0}^{n} \begin{cases} 1 & \text{if } l_i = 0 \text{ and } (T, t) \models \phi \\ 0 & \text{otherwise} \end{cases} \]

<table>
<thead>
<tr>
<th>Label</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>FN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FP</td>
<td>FP</td>
<td>TN</td>
</tr>
<tr>
<td>TN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem Formulation

Given a set of labeled signals, find a ptSTL formula describing the labeled events

\[ D = \{ (T, L) \mid T = (t_0, x_0), \ldots, (t_n, x_n), L = (t_0, l_0), \ldots, (t_n, l_n) \} \]

<table>
<thead>
<tr>
<th>Label</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>FN</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>FP</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TN</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Evaluation

\[ \text{Mismatch} \quad M = FP + FN \]
\[ \text{Precision} \quad P = \frac{TP}{TP + FP} \]
\[ \text{Recall} \quad R = \frac{TP}{TP + FN} \]
\[ \text{F beta} \quad F_b = \frac{(1+b^2) \cdot P \cdot R}{b^2 \cdot P + R} \]
Greedy Search for Formula Synthesis

Find $\phi$ minimizing cost

$$\phi = T \mid \phi_1 S_{[a,b]} \phi_2 \mid \phi_1 \land \phi_2 \mid \neg \phi \mid P_{[a,b]} \phi \mid A_{[a,b]} \phi \mid x_i \leq c \mid x_i \geq c$$

$$A_{[a,b]} x_i \geq c$$

$$P_{[a,b]} x_i \leq c$$

Parameter synthesis

$A_{[0,50]} x_1 \geq 20$

$P_{[0,20]} x_2 \leq 0$

Who will write the template formula?
Greedy Search for Formula Synthesis

Find $\phi$ minimizing cost

$\phi = T \mid \phi_1S_{[a,b]} \mid \phi_2 \mid \phi_1 \land \phi_2 \mid \neg \phi \mid P_{[a,b]} \phi \mid A_{[a,b]} \phi \mid x^i \leq c \mid x^i \geq c$

$\mathcal{F}^0 = \{x^i \sim p_i \mid \sim \in \{<, >\}, i = 1,\ldots,N\} \cup \{T\}$

$\mathcal{F}^n = \{\neg \phi \mid \phi \in \mathcal{F}^{n-1}\}$

$\mathcal{F}^{\leq n} = \bigcup_{i=0}^{n} \mathcal{F}^i$

$\mathcal{F}^n = \bigcup_{i=1}^{n-1} \{\phi_1 \land \phi_2 \mid \phi_1 \in \mathcal{F}^i, \phi_2 \in \mathcal{F}^{n-i-1}\}$

$\mathcal{F}^n = \bigcup_{i=1}^{n-1} \{\phi_1 \lor \phi_2 \mid \phi_1 \in \mathcal{F}^i, \phi_2 \in \mathcal{F}^{n-i-1}\}$

$\mathcal{F}^n = \bigcup_{i=1}^{n-1} \{\phi_1S_{[p_a,p_b]} \phi_2 \mid \phi_1 \in \mathcal{F}^i, \phi_2 \in \mathcal{F}^{n-i-1}\}$
Greedy Search for Formula Synthesis

Find $\phi$ minimizing cost

$$\mathcal{F} \leq n = \bigcup_{i=0}^{n} \mathcal{F}_i$$

Given an operator limit, run parameter optimization for each

$$\phi^p \in \mathcal{F} \leq n$$

$$\phi = T \mid \phi_1 S_{[a,b]} \phi_2 \mid \phi_1 \land \phi_2 \mid \neg \phi \mid P_{[a,b]} \phi \mid A_{[a,b]} \phi \mid x^i \leq c \mid x^i \geq c$$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c^i$</td>
<td>L$c^i$</td>
</tr>
<tr>
<td>$a$</td>
<td>L$a$</td>
</tr>
<tr>
<td>$b$</td>
<td>L$b$</td>
</tr>
</tbody>
</table>

Given a parametric formula $\phi^p$, and parameter limits, run grid search.
Greedy Search for Formula Synthesis - Example

\[ x^0, x^1, x^2, x^3, x^4, s^0, s^1 \]
\[ x^0, x^1, x^2 \in [0,40] \]
\[ x^3, x^4 \in [0,20] \]
\[ s^0, s^1 \in \{0,1\} \]

Simulate, assign label 1 when a link has more vehicles than \%75 of its capacity

\[ x^0 < 30 \land x^1 < 30 \land x^2 < 30 \land x^3 < 15 \land x^4 < 15 \]

Identify it, before it happens:

\[ P_{[1,1]}\phi^p, \phi^p \in F_{\leq n} \]

Operator count limit 0 (\(x^i \leftrightarrow c\)), 14 formulas.

\[ P_{[1,1]}s_0 = 0 \quad \text{fp: 307 fn: 256 tp: 599 tn: 638, f03: 0.664} \]

Operator count limit 1, 14 + 156 formulas.

\[ P_{[1,1]}(x^3 \geq 10.0 \land s^0 = 0) \quad \text{fp: 0 fn: 375 tp: 480 tn: 945, f03: 0.939} \]

Operator count limit 2, 14 + 156 + 5532 formulas.

\[ P_{[1,1]}(x^3 \geq 10.0 \land s^0 = 0) \quad \text{fp: 0 fn: 375 tp: 480 tn: 945, f03: 0.939} \]

Operator count limit 3, 14 + 156 + 5532 + 96414 formulas.

\[ P_{[1,1]}(x^4 \geq 10 \land s^1 = 0) \lor (x^3 \geq 10 \land s^0 = 0)) \quad \text{fp: 0 fn: 70 tp: 785 tn: 945, f03: 0.993} \]

DOES NOT SCALE
Greedy Search for Formula Synthesis - Example

\[ x^0, x^1, x^2, x^3, x^4, s^0, s^1 \]
\[ x^0, x^1, x^2 \in [0,40] \]
\[ x^3, x^4 \in [0,20] \]
\[ s^0, s^1 \in \{0,1\} \]

Minimize \( F 0.3 \)

Simulate, assign label 1 when a link has more vehicles than 75% of its capacity

\[ x^0 < 30 \land x^1 < 30 \land x^2 < 30 \land x^3 < 15 \land x^4 < 15 \]

\[ P_{[1,1]}((x^4 \geq 10 \land s^1 = 0) \lor (x^3 \geq 10 \land s^0 = 0)) \]

fp: 0 fn: 70 tp: 785 tn: 945, f03: 0.993

misclassification rate: 3.9 %

- There can be multiple causes.
- Increasing the operator count does not scale.
Iterative Formula Search-Heuristic 1

THE IDEA

- Given upper and lower bounds on operator count
- Found formulas: $\Phi^* = False$
- Start with $n=$lower bound, until $n >$ upper bound
  - For each $\phi^p \in \mathcal{F}^n$
    - Perform parameter optimization for $P_{[1,1]}(\phi^p \lor \Phi^*)$
    - Get the optimal one $\phi^*$
      - If there is sufficient improvement, update the found formula $\Phi^* = \Phi^* \lor \phi^*$
      - Otherwise increment $n$

1. $n=0$, best formula $P_{[1,1]}s_0 = 0$  $f03: 0.664$
   Insufficient, increment $n$

2. $n=1$, best formula $P_{[1,1]}(x^3 \geq 10.0 \land s^0 = 0)$  $f03: 0.939$

   Update $\Phi^* = P_{[1,1]}(x^3 \geq 10.0 \land s^0 = 0)$

3. $n=1$, best formula $P_{[1,1]}(x^4 \geq 10.0 \land s^1 = 0)$  $f03: 0.993$

   Update $\Phi^* = P_{[1,1]}((x^3 \geq 10.0 \land s^0 = 0) \lor (x^4 \geq 10.0 \land s^1 = 0))$

4. $n=1$, best formula $P_{[1,1]}(x^0 \geq 25 \land s^1 = 1)$  $f03: 0.995$

   Insufficient, increment $n$

$fp: 0$  $fn: 70$  $tp: 785$  $tn: 945$
Iterative Formula Search-Heuristic 2

THE IDEA

• Given upper bound on operator count \( n \)
• Found formulas: \( \Phi^* = False \)
• For each \( \phi^p \in \mathcal{F}^n \)
  • Perform parameter optimization for \( P_{[1,1]}\phi^p \)
• While the sub-formula limit is not exceeded
  • Select \( \phi^* \) maximizing \( TP(\Phi^* \lor \phi^*) \) and set \( \Phi^* = \Phi^* \lor \phi^* \)
  • Stop when the improvement is insufficient

1 First iteration \( \phi_1 = P_{[1,1]}(x^1 > 15 \land s^0 = 0 \land s^1 = 1) \)
2 Second iteration \( \phi_2 = P_{[1,1]}(x^1 \geq 25 \land s^1 = 1) \)
3 Third iteration \( \phi_3 = P_{[1,1]}(x^4 \leq 10 \land s^0 = 0 \land s^1 = 1) \)

\( \phi_1 \lor \phi_2 \lor \phi_3 \)

\( fp: 30 \; fn: 2 \; tp: 454 \; tn: 1314, \; mr = 1.8\% \)

There can be multiple causes.
Increasing the operator count does not scale.
• Iterative cause finding?
When the aircraft’s longitudinal motion disturbs?

\[ \alpha^0 : \text{no wind} \]
\[ \alpha^1 : \text{wind (random)} \]
Iterative Formula Search-Heuristic 2 - Example

When the aircraft’s longitudinal motion disturbs?

**alpha**

1. \( \phi_1 = P_{[4,10]}(q\text{Gust} \leq 0 \land w\text{Gust} \leq -120) \)
2. \( \phi_2 = w\text{Gust} \geq 120 \land A_{[14,14]}(\text{pilot} \geq -0.4) \)
3. \( \phi_3 = P_{[2,2]}(\alpha \leq 0.3 \land w\text{Gust} - \leq 120) \)
4. \( \phi_4 = (A_{[2,6]}q\text{Gust} \geq 0.1) \land \text{pilot} \leq -0.4 \)

\( \phi_1 \lor \phi_2 \lor \phi_3 \lor \phi_4 \)  

fp: 18 fn: 58 tp: 419 tn: 2505,  mr=2.5%
Controllable Formula Search (ongoing work)

Find the cause and update the system to avoid it in fully automated way.

\[ \Phi_1 \lor \ldots \lor \Phi_p \]

\[ \Phi_i := (A_{[1,b_i]}u^j = c_i) \land (P_{[1,1]}\phi_i) \]

Controller: Select \( u \) that violates each cause

ITERATIVELY

Find cause
Synthesize controller
Simulate-label
Controllable Formula Search (ongoing work)

\[ \Phi_1 \lor \ldots \lor \Phi_p \]
\[ \Phi_i := (A_{[1,b_i]}^i u^i = c_i) \land (P_{[1,1]} \phi_i) \]

<table>
<thead>
<tr>
<th>i</th>
<th>Viol Count</th>
<th>( A_{[1,2]}(s^0 = 1) \land P_{[1,1]}(x^0 &gt; 20) )</th>
<th>tp</th>
<th>fp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1591</td>
<td>405</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1549</td>
<td>576</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1557</td>
<td>465</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>1763</td>
<td>599</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Controllable Formula Search (ongoing work)

Φ₁ ∨ ... ∨ Φₚ
Φᵢ := (A₁,bᵢ)uᵢ = cᵢ) ∧ (P₁,₁)φᵢ

<table>
<thead>
<tr>
<th>Ex#</th>
<th>time</th>
<th>v₀</th>
<th>v₁</th>
<th>v₂</th>
<th>v₃</th>
<th>v₄</th>
<th>v₅</th>
<th>v₆</th>
<th>v₇</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>630sec</td>
<td>143</td>
<td>23</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>#2</td>
<td>320sec</td>
<td>911</td>
<td>530</td>
<td>265</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>#3</td>
<td>115sec</td>
<td>911</td>
<td>617</td>
<td>149</td>
<td>80</td>
<td>70</td>
<td>48</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ex#</th>
<th>oc</th>
<th>oc</th>
<th>ñ</th>
<th>val</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>0.01</td>
</tr>
<tr>
<td>#2</td>
<td>0</td>
<td>∞</td>
<td>∞</td>
<td>0.1</td>
</tr>
<tr>
<td>#3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

No formal guarantees, but much faster compared to[1], and applicable to any system

Controllable Formula Search - Robotic Example

\[ \Phi_1 \lor \ldots \lor \Phi_p \]
\[ \Phi_i := (A_{[1,b,i]}u^i = c_i) \land (P_{[1,1]}\phi_i) \]

Random walk, label - 1 when hits an obstacle

Control actions: S,N,E,W

<table>
<thead>
<tr>
<th>i</th>
<th>vi</th>
<th>( \Phi_i )</th>
<th>( t_p )</th>
<th>( f_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>705</td>
<td>( (G_{[1,1]}^- u = S) \land (F_{[1,1]}^- (x^1 &gt; 3 \land x^0 &gt; 2)) )</td>
<td>126</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>429</td>
<td>( (G_{[1,1]}^- u = N) \land (F_{[1,1]}^- (x^1 &lt; 4 \land x^0 &lt; 3)) )</td>
<td>125</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>109</td>
<td>( (G_{[1,1]}^- u = E) \land (F_{[1,1]}^- (x^1 &gt; 2 \land x^0 &gt; 3)) )</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>77</td>
<td>( (G_{[1,1]}^- u = W) \land (F_{[1,1]}^- (x^1 &lt; 5 \land x^0 &lt; 2)) )</td>
<td>52</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
Summary: Cause mining with STL

$\phi_1 = P_{[1,1]}(x^1 > 15 \land s^0 = 0 \land s^1 = 1)$

$\phi_2 = P_{[1,1]}(x^1 \geq 25 \land s^1 = 1)$

$\phi_3 = P_{[1,1]}(x^4 \leq 10 \land s^0 = 0 \land s^1 = 1)$

$x_i[t + 1] = x_i[t] + d_i[t] - f_i^{out}(x_i[t]) + \sum_{k \in C_i^{out}} \beta_k f_k^{out}(x_k[t])$
Cause mining with STL

- System model
- Simulations
- Label

Identify the patterns leading the labeled events as STL formulas

- How can we ensure that these are actual causes?
  - Underlying causes?
  - Symptoms?

- What if there is single label?
  - Somewhere along the simulation a problem occurred.

- Is there a class of systems for which this type of analysis can reveal the causes?

Further questions?