Causal Analysis of Probabilistic Counterexamples

Hichem Debbi  
hichem.debbi@gmail.com

Mustapha Bourahla  
mbourahla@hotmail.com

University of M’Sila
Motivation

Counterexample Analysis

- Inevitable complementary task to counterexample generation
- Error location is the most difficult part of debugging [Vesey]

Debugging Probabilistic Models

To answer the question:
Why is the probability threshold violated?

Challenges for Analysing Probabilistic Counterexamples

- Multiple Paths
- Probabilistic Nature
Probabilistic Computation Tree Logic

**PCTL Logic**

PCTL is an extension of CTL for specifying probabilistic properties.

**State Formula**

\[ \phi ::= \text{true} | a | \neg \phi | \phi_1 \land \phi_2 | P_{\sim p}(\varphi) \]

**Path Formula**

\[ \varphi ::= \phi_1 U \phi_2 | \phi_1 W \phi_2 | \phi_1 U^{\leq n} \phi_2 | \phi_1 W^{\leq n} \phi_2 \]

**PCTL Property Satisfaction**

\[ s \models P_{\sim p}(\varphi) \iff P(s \models \varphi) \sim p \]

\[ Pr(s \models \varphi) = Pr_s \{ \pi \in Paths(s) | \pi \models \varphi \} \sim \in \{<, \leq, >, \geq\} \]
Probabilistic Counterexamples

A counterexample $C$ for $P_{\leq p}(\varphi)$ is a set of finite paths with $\Pr(C) > p$

$s \not\models P_{\leq 0.01}(F \text{ error})$ \quad \Rightarrow \quad \Pr(C) > 0.01$
\[ P_{\leq 0.5}[(a \lor b)U(c \land d)] \]

\[
P(CX_2) = P \left( \{s_0s_1, s_0s_2s_3, s_0s_2s_4s_3, s_0s_2s_4s_5, s_0s_4s_5 \} \right)
= 0.25 + 0.2 + 0.09 + 0.15 + 0.12 = 0.81
\]
\[ P_{\leq 0.5}[(a \lor b) U (c \land d)] \]

\[ P(CX_2) = P(\{s_0s_1, s_0s_2s_3, s_0s_2s_4s_3, s_0s_2s_4s_5, s_0s_4s_5 \}) \]

\[ = 0.25 + 0.2 + 0.09 + 0.15 + 0.12 = 0.52 \]
\[ P_{\leq 0.5}[(a \lor b)U(c \land d)] \]

Most Indicative

\[ P(CX_2) = P(\{s_0s_1, s_0s_2s_3, s_0s_2s_4s_3, s_0s_2s_4s_5, s_0s_4s_5\}) = 0.25 + 0.2 + 0.09 + 0.15 + 0.12 = 0.60 \]
Find Labeling and probability values in the counterexample that cause the probability to exceed the given upper bound over the model.
### Causality and Responsibility for MIPCX

#### Criticality

$(s, X = x)$ is critical if \(\overline{MIPCX}_{(s,X=x)}(s_0 \models \Phi)\) is not a valid counterexample.

\[
\overline{MIPCX}_{(s,X=x)}(s_0 \models \Phi) :
\]

The set of finite paths resulting from \(MIPCX(s_0 \models \Phi)\) by switching the value of variable $X$ in state $s$

#### Causality (adapted from Halpern & Pearl)

$(s, X = x)$ is a cause for violating MIPCX if either $(s, X = x)$ is critical or $W \leftarrow w'$ makes $(s, X = x)$ critical, for variable subset $W$

#### Degree of Responsibility (adapted from Chockler & Halpern)

\[
dR(s, X = x, \Phi) = \begin{cases} 1 & \text{if } (s, X = x) \text{ is critical} \\ 1/(|W| + 1) & \text{otherwise} \end{cases}
\]
Probabilistic Causality Model

is a tuple $< M, Pr >$

$M$ : causality model and

$Pr$ : probability function defined over the states of $MIPCX(s_0 \models \Phi)$

$$Pr(s) = \sum_{\sigma \in \sigma \mid \sigma \in MIPCX(s_0 \models \Phi)} P(\sigma) \quad Pr(s, X = x) = Pr(s)$$

Most Responsible Cause

Cause $C$ is a most responsible cause for violating $\Phi = P_{\leq p}(\varphi)$ if $dR(C)Pr(C) \geq dR(C')Pr(C')$ for any cause $C'$. 
$P_{\leq 0.5}[(a \lor b) \cup (c \land d)]$

Most Indicative

$P(CX_2) = P(\{s_0s_1, s_0s_2s_3, s_0s_2s_4s_3, s_0s_2s_4s_5, s_0s_4s_5\})$

$= 0.25 + 0.2 + 0.09 + 0.15 + 0.12 = 0.60$
Probabilistic Counterexamples Revisited

\[ P_{\leq 0.5}[(a \lor b) U (c \land d)] \]

\( dR(s_4, b = 1) = 1/|\{a\}| + 1 = 0.5 \)

\( dR(s_2, b = 1) = 1 \)

\( Pr(s_2, b = 1) = 0.2 + 0.15 = 0.35 \)

\( dR(s_2, b = 1)Pr(s_2, b = 1) = 0.35 \) : highest

\( (s_2, b=1) \) is the most responsible cause
Algorithm and Implementation

Probabilistic Symbolic Model Checker
[Kwiatkowska et al.]

Probabilistic Counterexample Generator
[Aljazzar et al.]

\[ MIPC(s_0 = \Phi) \quad \Phi = P_{\leq p}(\varphi) \]

Debugging Algorithm (Debbi-Bourahla)

Diagnosis

Causes with Responsibilities and Probabilities
## Conclusion

- We adapted and showed the usefulness of Causality and Responsibility in the context of debugging probabilistic counterexamples.
- We introduced the notion of Most Responsible Cause as an indicator for the source of the error.
- We developed a Debugging Algorithm, and tested it on real case studies with good performance.

## Future Work

- Visualization of diagnosis results.