Falsification of Conditional Safety Properties for Cyber-Physical Systems with Gaussian Process Regression

Takumi Akazaki(✉)

The University of Tokyo, Tokyo, Japan
ultraredrays@is.s.u-tokyo.ac.jp

Abstract. We propose a framework to solve falsification problems of *conditional safety properties*—specifications such that "a safety property φ_{safe} holds whenever an antecedent condition φ_{cond} holds." In the outline, our framework follows the existing one based on *robust semantics* and numerical optimization. That is, we search for a counterexample input by iterating the following procedure: (1) pick up an input; (2) test how robustly the specification is satisfied under the current input; and (3) pick up a new input again hopefully with a smaller robustness. In falsification of conditional safety properties, one of the problems of the existing algorithm is the following: we sometimes iteratively pick up inputs that do not satisfy the antecedent condition φ_{cond}, and the corresponding tests become less informative. To overcome this problem, we employ *Gaussian process regression*—one of the model estimation techniques—and estimate the region of the input search space in which the antecedent condition φ_{cond} holds with high probability.

1 Introduction

1.1 Falsification

In design of Cyber-Physical Systems (CPSs), the importance of quality assurance of these systems is ever-rising, thus employing *model-based development (MBD)*—making virtual models (e.g. Simulink/Stateflow blocks) of products, and on these models, verifying properties by mathematical methodologies—has become standard. However, currently at least, the complexity of these virtual models in industry are overwhelm the scalability of the state-of-art formal verification methodologies.

Under such current situation, *falsification* is gathering attention as a viable approach to quality assurance [2,6,9,11]. The falsification problem is formulated as follows.

- **Given:** a system *model* $\mathcal{M}$ with its input domain D, and a *specification* φ
- **Return:** a counterexample input $x \in D$ such that its corresponding output $\mathcal{M}(x)$ violates the specification φ (if such an input exists).

Y. Falcone and C. Sanchez (Eds.): RV 2016, LNCS 10012, pp. 439–446, 2016.
DOI: 10.1007/978-3-319-46982-9_27

Through solving the above falsification problem, we expect to obtain the following insights: (1): we detect errors in which the system violates the specification φ; and (2): in case that such an error could not be found, we would say "the violation of the specification φ unlikely happens."

1.2 Robustness Guided Falsification

As a formal expression of real-time specification on CPSs, *metric interval temporal logic (**MITL**)* [1], and its adaptation *signal temporal logic (**STL**)* [12] are actively studied. For these specifications, one common class of algorithms to solve falsification is *robustness guided falsification* [2,6]. Here, one technical core of these algorithms is employing *robust semantics* [7,8] on these logics. In robust semantics, in contrast to conventional Boolean semantics, a truth value takes a quantitative one $[\![\mathcal{M}(x),\ \varphi]\!] \in \mathbb{R}$ such that it is greater than 0 if the formula φ is satisfied, and its magnitude denotes "how robustly the current output $\mathcal{M}(x)$ satisfies φ." With this robust semantics, we could attribute falsification problems to numerical optimization problems, that is, we search for a counterexample input $x \in D$ by iterating the following steps (for $t = 1 \ldots N$).

1. Pick an input $x_t \in D$ (in stochastic manner.)
2. Compute the output $\mathcal{M}(x_t)$ by numerical simulation (e.g. sim function on Simulink)
3. Check the robustness $[\![\mathcal{M}(x_t),\ \varphi]\!]$
4. If the robustness is less equal than 0, then return x_t. Otherwise pick a new input x_{t+1} hopefully with which the robustness becomes smaller.

In industrial practice, a system model $\mathcal{M}$ is often huge and complex, hence among the above four steps, the second one, numerical simulation step tends to be the most costly in time—it sometimes takes several tens of seconds for each simulation. Therefore, reducing the number of iterations in minimization of the robustness $[\![\mathcal{M}(x_t),\ \varphi]\!]$ is essential. To this end, application of various numerical optimization algorithms (e.g. Simulated Annealing [2], Cross-entropy method [14], and so on) is actively studied.

In this paper, as one of the powerful numerical optimization algorithms, we mainly employ *Gaussian process upper confidence bound (**GPU-CB**)* [15,16]. Actually, applying **GP-UCB** and other Gaussian process regression based optimization techniques for falsification of temporal logic properties is actively studied. [3–5] We give further illustration of **GP-UCB** in Sect. 3.

1.3 Our Motivation: Falsification of Conditional Safety Property

In this paper, as a class of specifications to be falsified, we have an eye on *conditional safety properties*—common class of specifications in development of CPSs.

> Whenever a model satisfies an antecedent condition φ_{cond}, then at that time, the model also satisfies a safety property φ_{safe}.

With this class of formulas, we could express various requirements of behavior of the system under various specific conditions. Hence, for a given system, verifying conditional safety property is as important as for safety property.

On **STL**, we usually encode such a condition into a **STL** formula in the form of $\Box_I(\neg\varphi_{\mathsf{cond}} \vee \varphi_{\mathsf{safe}})$. Note that, in conventional Boolean semantics, the formula is equivalent to $\Box_I(\varphi_{\mathsf{cond}} \rightarrow \varphi_{\mathsf{safe}})$. In robustness guided falsification, we search for a counterexample by minimizing the robustness of the formula $\neg\varphi_{\mathsf{cond}}$ and φ_{safe} simultaneously.

However there exists the following gap between this straightforward attribution to the numerical optimization and what we expect to obtain through the falsification: if we write down a conditional safety property, we would like to say something meaningful about dynamics of the model when the antecedent condition φ_{cond} holds; but in the iteration of simulation, *we could not guarantee that enough number of behavior are observed in which the system satisfies the antecedent condition* φ_{cond}. From this point of view, we would expect an optimization algorithm that solves conditional safety property

- with as small as number of iteration to find a counterexample $x \in D$; and
- with picking up enough number of inputs $x_{j_1} \dots x_{j_n}$ that steers the whole model to satisfy the antecedent condition φ_{cond}.

To this end, we propose a novel algorithm to pick up a suitable input in each step of the iteration with satisfying the above twofold requirements. A technical highlight is that, with Gaussian process regression, we estimate the function $F^* : x \mapsto [\![\mathcal{M}(x), \Box_I\neg\varphi_{\mathsf{cond}}]\!]$, and obtaining the input subspace $D' \subset D$ such that, for any input $x \in D'$, the output $\mathcal{M}(x)$ satisfies the antecedent condition φ_{cond} with high probability.

Related Work. The difficulty of the falsification is to observe the rare event (here, conditional safety property is false). Our technique is based on the following idea: we consider a superset-event that happens much likely than the original one (φ_{cond} holds), and from the input space, we "prune" the region in which the superset-event does not happen. This idea is common with importance sampling. Actually, our Proposition 2.4 is an instance of decomposition in Sect. 4.1 in [10].

While importance sampling explores the input by stochastic sampling, **GP-UCB** deterministically chooses the next input, hence combining these two optimization algorithms are not straightforward. One of our contributions is that we realize the above "pruning" in GP-UCB style optimization by employing regression.

2 Signal Temporal Logic (STL)

Definition 2.1 (syntax). Let **Var** be a set of variables. The set of **STL** *formulas* are inductively defined as follows.

$$\varphi ::= f(v_1, \dots, v_n) > 0 \mid \bot \mid \top \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi\,\mathcal{U}_I\,\varphi$$

where f is an n-ary function $f : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, \infty\}$, $v_1, \ldots, v_x \in$ **Var**, and I is a closed non-singular interval in $\mathbb{R}_{\geq 0}$, i.e. $I = [a, b]$ or $[a, \infty)$ where $a < b$ and $a \in \mathbb{R}$. We also define the following derived operators, as usual: $\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2)$, $\varphi_1 \,\mathcal{R}_I\, \varphi_2 \equiv \neg(\neg\varphi_1 \,\mathcal{U}_I\, \neg\varphi_2)$, $\Diamond_I \varphi \equiv \top \,\mathcal{U}_I\, \varphi$, and $\Box_I \varphi \equiv \bot \,\mathcal{R}_I\, \varphi$.

Definition 2.2 (robust semantics of STL). Let $\sigma \colon \mathbb{R}_{\geq 0} \to \mathbb{R}^{\mathbf{Var}}$ be a signal and φ be an **STL** formula. We define the *robustness* $\llbracket \sigma,\, \varphi \rrbracket \in \mathbb{R}_{\geq 0} \cup \{-\infty, \infty\}$ inductively as follows. Here $\sqcap$ and $\sqcup$ denote infimums and supremums of real numbers, respectively.

$$\llbracket \sigma,\, f(v_1, \cdots, v_n) > 0 \rrbracket \triangleq f\big(\sigma(0)(v_1), \cdots, \sigma(0)(v_n)\big)$$

$$\llbracket \sigma,\, \bot \rrbracket \triangleq -\infty \qquad\qquad \llbracket \sigma,\, \top \rrbracket \triangleq \infty$$

$$\llbracket \sigma,\, \neg\varphi \rrbracket \triangleq -\llbracket \sigma,\, \varphi \rrbracket \qquad\qquad \llbracket \sigma,\, \varphi_1 \vee \varphi_2 \rrbracket \triangleq \llbracket \sigma,\, \varphi_1 \rrbracket \sqcup \llbracket \sigma,\, \varphi_2 \rrbracket$$

$$\llbracket \sigma,\, \varphi_1 \,\mathcal{U}_I\, \varphi_2 \rrbracket \triangleq \bigsqcup_{t \in I} \Big(\llbracket \sigma^t,\, \varphi_2 \rrbracket \sqcap \bigsqcap_{t' \in [0,t)} \llbracket \sigma^{t'},\, \varphi_1 \rrbracket \Big)$$

Notation 2.3. Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty, \infty\}$. We define *the Boolean abstraction of f* as the function $\overline{f} : \mathbb{R}^n \to \mathbb{B}]$ such that as $\overline{f}(v) = \top$ if $f(v) > 0$, otherwise $\overline{f}(v) = \bot$. Similarly, for an **STL** formula φ, we denote by $\overline{\varphi}$ the formula which is obtained by replacing all atomic functions f occurs in φ with the Boolean abstraction $\overline{f}$. We see that $\llbracket \sigma,\, \varphi \rrbracket > 0$ implies $\llbracket \sigma,\, \overline{\varphi} \rrbracket > 0$.

As we see in Sect. 1.3, conditional safety properties are written as **STL** formulas in the form of $\llbracket \sigma,\, \Box_I(\neg\varphi_{\mathsf{cond}} \vee \varphi_{\mathsf{safe}}) \rrbracket$, and its intuitive meaning is "φ_{safe} holds whenever φ_{cond} is satisfied." To enforce our algorithm in Sect. 4 to pick inputs satisfying the antecedent conditions φ_{cond}, we convert the formula to the logically equivalent one. The converted formula consists of mainly into the two parts such that one of them stands for "the antecedent condition φ_{cond} is satisfied or not."

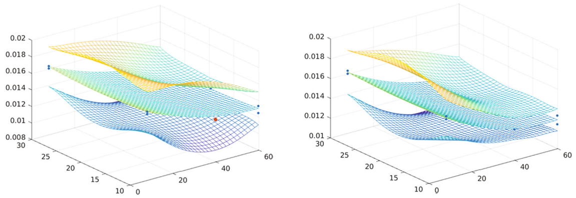
Proposition 2.4. *For any signal σ and **STL** formulas φ_1, φ_2, the following holds.*

$$\llbracket \sigma,\, \Box_I(\neg\varphi_1 \vee \varphi_2) \rrbracket > 0 \iff \llbracket \sigma,\, \Box_I \neg\varphi_1 \rrbracket \sqcup \llbracket \sigma,\, \Box_I(\neg\overline{\varphi_1} \vee \varphi_2) \rrbracket > 0$$

3 Gaussian Process Upper Confidence Bound (GP-UCB)

As we mentioned in Sect. 1.3, in robustness guided falsification to minimize $F^* : x \mapsto \llbracket \mathcal{M},\, \varphi \rrbracket$, we pick inputs iteratively hopefully with smaller robustness value. For this purpose, *Gaussian process upper confidence bound (**GP-UCB**)* [15,16] is one of the powerful algorithm as we see in [3–5].

The key idea in the algorithm is that, in each iteration round $t = 1, \ldots, N$, we estimate the *Gaussian process* [13] $\mathrm{GP}(\mu, k)$ that most likely to generate the points observed until round t. Here, we call two parameters $\mu : D \to \mathbb{R}$ and $k : D^2 \to \mathbb{R}$ as the *mean function* and the *covariance function* respectively.



at iteration t at iteration $t + 1$

Fig. 1. An intuitive illustration of **GP-UCB** algorithm. Each figure shows the estimated Gaussian process $\mathrm{GP}(\mu, k)$ at iteration round t and $t + 1$: the middle curve is a plot of the mean function μ, and the upper and lower curve are a plot of $\mu + \beta^{1/2}k$, $\mu - \beta^{1/2}k$. In each iteration round t, we pick the point $x[t]$ (red point in the left figure) that minimizes the lower curve. Once we observe the value $F^*(x[t])$, the uncertainty at $x[t]$ becomes smaller in the next round $t + 1$. In general, as a confidence parameter β we choose an increasing function to guarantee the algorithm not to get stuck in local optima (e.g. $\beta(t) = 2\log(ct^2)$ for some constant c). See [15,16]) (Color figure online)

Very roughly speaking, for each $x \in D$, the value $\mu(x)$ of mean function stands for the expected value of $F^*(x)$, and the value $k(x, x)$ of co variance function at each diagonal point does for the magnitude of uncertainty of $F^*(x)$.

Pseudocode for the **GP-UCB** algorithm is found in Algorithm 1. As we see in the code, we pick $x[t] = \mathsf{argmin}_{x \in D}\, \mu(x) - \beta^{1/2}(t)k(x, x)$ as the next input. Here, the first term try to minimize the expected value $F^*(x[t])$, and the second term try to decrease uncertainty globally. In Fig. 1, we see an illustration of how the estimated Gaussian process is updated in each iteration round of optimization. Thus, the strategy balancing exploration and exploitation helps us to find a minimal input with as small as number of iteration.

Algorithm 1. The GP-UCB algorithm for falsification

```
Hyper parameters: A confidence parameter $\beta : \mathbb{N} \to \mathbb{R}$; Maximal number of iteration $N$;
Input: Input space $D$; An uncertain function $F : D \to \mathbb{R}$ to be minimized;
Output: An input $x \in D$ such that $F(x) \leq 0$
  for $t = 1 \ldots N$ do
     $x[t] = \mathsf{argmin}_{x \in D}\, \mu(x) - \beta^{1/2}(t)k(x, x)$;          ▷ Choose a new sample input
     $y[t] = F(x[t])$;                                         ▷ Observe the corresponding output
     if $y[t] \leq 0$ then
          return $x[t]$;
     end if
     $(\mu, k) = \mathsf{regression}\big((x[1], y[1]), \ldots (x[t], y[t])\big)$;
                         ▷ Perform Bayesian update to obtain new mean and covariance function
  end for
```

4 Our Algorithm: GP-UCB with Domain Estimation

Now we give our algorithm for falsification of conditional safety properties with enough number of testing in which the model satisfies the antecedent condition.

Algorithm 2. The GP-UCB algorithm for falsification with domain estimation

```
Hyper parameters: A confidence parameter $\beta : \mathbb{N} \to \mathbb{R}$ and its bound $\beta_{\mathsf{min}}, \beta_{\mathsf{max}} \in \mathbb{R}$; Maximal
   number of iteration $N$; Target hit rate $R \in (0, 1)$
Input: Input space $D$; Uncertain functions $F, G : D \to \mathbb{R}$;
Output: An input $x \in D$ such that $\mathsf{max}(F(x), G(x)) \leq 0$
   for $t = 1 \ldots N$ do
      $r = (R \times N - n_{\mathsf{hit}})/(N - t)$
                              ▷ Calculate the current objective probability $r$ of satisfying $F(x) \leq 0$
      $\beta_F = \mathsf{min}(\mathsf{max}(\sqrt{2}\mathsf{erf}^{-1}(1 - 2r), \beta_{\mathsf{min}}), \beta_{\mathsf{max}})$ where erf is the error function
      $D' = \{x \in D \mid \mu_F(x) - \beta_F k_F(x, x) \leq 0\}$
                                    ▷ Estimate a region in which $F(x) \leq 0$ holds with probability $r$
      if $D' == \emptyset$ then
         $x_F[t] = \mathsf{argmin}_{x \in D}\, \mu_F(x) - \beta_F k_F(x, x)$;
      else
         $x_G[t] = \mathsf{argmin}_{x \in D'}\, \mu_G(x) - \beta^{1/2}(t) k_G(x, x)$;
      end if                                                  ▷ Choose a new sample input
      $y_F[t] = F(x_t)$;
      if $y_F[t] \leq 0$ then
         $n = n + 1$; $x_G[n] = x_F[t]$; $y_G(x_G[n])$;
         if $y_G[n] \leq 0$ then
            return $x_G[n]$;
         end if
      end if                                           ▷ Observe the corresponding output
      $(\mu_F, k_F) = \mathsf{regression}\big((x_F[1], y_F[1]), \ldots (x_F[t], y_F[t])\big)$;
      $(\mu_G, k_G) = \mathsf{regression}\big((x_G[1], y_G[1]), \ldots (x_G[n], y_G[n])\big)$;
                           ▷ Perform Bayesian update to obtain new mean and covariance function
   end for
```

As we show in Proposition 2.4, falsification of the specification $\Box_I(\neg\varphi_{\mathsf{cond}} \vee \varphi_{\mathsf{safe}})$ could be reduced to the following problem.

$$\text{Find } x \text{ such that } [\![\mathcal{M}(x), \Box_I \neg\varphi_{\mathsf{cond}}]\!] \sqcup [\![\mathcal{M}(x), \Box_I(\neg\overline{\varphi_{\mathsf{cond}}} \vee \varphi_{\mathsf{safe}})]\!] \leq 0.$$

A key observation here is that, when the first part of the robustness $[\![\mathcal{M}(x), \Box_I \neg\varphi_{\mathsf{cond}}]\!]$ becomes less than zero, then with this input x, the corresponding behavior of the system $\mathcal{M}(x)$ satisfies the antecedent condition φ_{cond}.

Based on this observation, we propose the **GP-UCB** with domain estimation algorithm. Pseudocode of the algorithm is available in Algorithm 4. This algorithm takes a hyper parameter R which stands for a target hit rate, that is, how large ratio of the input $x[1], ..., x[N]$ steer the model to satisfy the antecedent condition. In each iteration round of the falsification, to guarantee both fast minimization and enough testing on which φ_{cond} holds, we pick the next input by the following strategy: (1) calculate how many ratio r of the input should make φ_{cond} true through the remaining iteration; (2) estimate the input subdomain $D' \subset D$ in which the antecedent condition φ_{cond} holds with probability r; (3) from the restricted domain $x \in D'$, pick a new input x to falsify the whole specification in the **GP-UCB** manner.

5 Experiments

To examine that our **GP-UCB** with domain estimation algorithm achieves both fast minimization and enough testing with the antecedent condition φ_{cond}.

As a model of the CPSs, we choose the powertrain control verification benchmark [11]. This is an engine model with a controller which try to keep the air/fuel ratio in the exhaust gas. This model has 3-dimensional input parameters, and the controller have mainly two modes—feedback mode and feed-forward mode. As conditional safety specifications to falsify, we experiment with the following **STL** formula φ. In this formula, the antecedent condition is $\mathsf{mode} = \mathsf{feedforward}$, that is, we would like to observe behavior of the system in the feed-forward mode.

$$\Box_{[\tau,\infty)}\big(\neg(\mathsf{mode} = \mathsf{feedforward}) \vee |\mathsf{ratio}_{\mathsf{A/F}}| < 0.2\big) \qquad (1)$$

In fact of the model, the formula (1) does not have any counterexample input, and with the original **GP-UCB** algorithm, about 58 % of the input leads the whole systems to feed-forward mode. Then, we run our **GP-UCB** with domain estimation algorithm with setting the target hit rate as $R = 0.8$, and observe that about 79 % of the inputs satisfy the antecedent condition.

6 Conclusion

To solve falsification of conditional safety properties with enforcing the generated inputs to satisfy the antecedent condition, we provide an optimization algorithm based on Gaussian process regression techniques.

References

1. Alur, R., Feder, T., Henzinger, T.A.: The benefits of relaxing punctuality. J. ACM **43**(1), 116–146 (1996)
2. Annpureddy, Y., Liu, C., Fainekos, G., Sankaranarayanan, S.: S-TaLiRo: a tool for temporal logic falsification for hybrid systems. In: Abdulla, P.A., Leino, K.R.M. (eds.) TACAS 2011. LNCS, vol. 6605, pp. 254–257. Springer, Heidelberg (2011)
3. Bartocci, E., Bortolussi, L., Nenzi, L., Sanguinetti, G.: On the robustness of temporal properties for stochastic models. In: Dang, T., Piazza, C. (eds.) Proceedings Second International Workshop on Hybrid Systems and Biology, HSB 2013, Taormina, Italy, 2nd September 2013, vol. 125 of EPTCS, pp. 3–19 (2013)
4. Bartocci, E., Bortolussi, L., Nenzi, L., Sanguinetti, G.: System design of stochastic models using robustness of temporal properties. Theor. Comput. Sci. **587**, 3–25 (2015)
5. Chen, G., Sabato, Z., Kong, Z.: Active requirement mining of bounded-time temporal properties of cyber-physical systems. CoRR abs/1603.00814 (2016)
6. Donzé, A.: Breach, a toolbox for verification and parameter synthesis of hybrid systems. In: Touili, T., Cook, B., Jackson, P. (eds.) CAV 2010. LNCS, vol. 6174, pp. 167–170. Springer, Heidelberg (2010)
7. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: Chatterjee, K., Henzinger, T.A. (eds.) FORMATS 2010. LNCS, vol. 6246, pp. 92–106. Springer, Heidelberg (2010)
8. Fainekos, G.E., Pappas, G.J.: Robustness of temporal logic specifications for continuous-time signals. Theor. Comput. Sci. **410**(42), 4262–4291 (2009)
9. Hoxha, B., Abbas, H., Fainekos, G.: Benchmarks for temporal logic requirements for automotive systems. In: Proceedings of Applied Verification for Continuous and Hybrid Systems (2014)

10. Jegourel, C., Legay, A., Sedwards, S.: Importance splitting for statistical model checking rare properties. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 576–591. Springer, Heidelberg (2013)
11. Jin, X., Deshmukh, J.V., Kapinski, J., Ueda, K., Butts, K.: Powertrain control verification benchmark. In: Fränzle, M., Lygeros, J. (eds.) 17th International Conference on Hybrid Systems: Computation and Control (part of CPS Week), HSCC 2014, Berlin, Germany, 15–17 April 2014, pp. 253–262. ACM (2011)
12. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Lakhnech, Y., Yovine, S. (eds.) FORMATS 2004 and FTRTFT 2004. LNCS, vol. 3253, pp. 152–166. Springer, Heidelberg (2004)
13. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, Massachusetts (2005)
14. Sankaranarayanan, S., Fainekos, G.: Falsification of temporal properties of hybrid systems using the cross-entropy method. In: Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2012, pp. 125–134. ACM, New York (2012)
15. Srinivas, N., Krause, A., Kakade, S., Seeger, M.W.: Gaussian process optimization in the bandit setting: no regret and experimental design. In: Fürnkranz, J., Joachims, T. (eds.) Proceedings of the 27th International Conference on Machine Learning (ICML 2010), 21–24 June 2010, Haifa, Israel, pp. 1015–1022. Omnipress (2010)
16. Srinivas, N., Krause, A., Kakade, S.M., Seeger, M.W.: Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. IEEE Trans. Inf. Theor. **58**(5), 3250–3265 (2012)