Safe Inputs Approximation for Black-Box Systems

Bai Xue\textsuperscript{1,2,3} Yang Liu\textsuperscript{4} Lei Ma\textsuperscript{5} Xiyue Zhang\textsuperscript{6} Meng Sun\textsuperscript{6} and Xiaofei Xie\textsuperscript{4}

\textsuperscript{1}State Key Lab. of Computer Science, Institute of Software, CAS, China, xuebai@ios.ac.cn
\textsuperscript{2} University of Chinese Academy of Sciences, China
\textsuperscript{3} Beijing Institute of Control Engineering, Beijing, China
\textsuperscript{4} School of Computer Science and Engineering, Nanyang Technological University, Singapore, {yangliu,xfxie}@ntu.edu.sg
\textsuperscript{5} Information Science and Electrical Engineering, Kyushu University, Japan, malei@ait.kyushu-u.ac.jp
\textsuperscript{6} School of Mathematical Science, Peking University, China, sumn@pku.edu.cn; xiyuezhangpku@gmail.com

Abstract—Given a family of independent and identically distributed samples extracted from the input region and their corresponding outputs, in this paper we propose a method to under-approximate the set of safe inputs that lead the black-box system to respect a given safety specification. Our method falls within the framework of probably approximately correct (PAC) learning. The computed under-approximation comes with statistical soundness provided by the underlying PAC learning process. Such a set, which we call a PAC under-approximation, is obtained by computing a PAC model of the black-box system with respect to the specified safety specification. In our method, the PAC model is computed based on the scenario approach, which encodes as a linear program. The linear program is constructed based on the given family of input samples and their corresponding outputs. The size of the linear program does not depend on the dimensions of the state space of the black-box system, thus providing scalability. Moreover, the linear program does not depend on the internal mechanism of the black-box system, thus being applicable to systems that existing methods are not capable of dealing with. Some case studies demonstrate these properties, general performance and usefulness of our approach.

Index Terms—Black-Box Systems; Linear Programming; Probably Approximate Safety

I. INTRODUCTION

Safety-critical systems are those systems whose failure could result in loss of life or damage to the environment. Safety verification for these systems is extremely important. Formal methods, which are the process of verifying with mathematical rigor that a system behaves correctly and are a well-established branch in computer science\textsuperscript{2}, are widely recognized as an essential step in the design process of safety-critical systems.

In formal methods, abstraction is part of the process of developing a mathematical model that is a simplification or approximation of reality but that retains the properties of interest, e.g., \textsuperscript{[3]}. In physics, for example, it is customary to model a moving object as a point mass, and to ignore its shape. Model checking and theorem proving are two pillars of formal methods\textsuperscript{[11]}. Model checking verifies properties against a model to prove that the design conforms to the specifications\textsuperscript{[10]}. Theorem proving relies heavily on higher-order logic and leverages mathematical structures to build formulas that correspond to the behavior of the system\textsuperscript{[31]}. The verification process, then, is the evaluation of these formulas. However, formal methods deal with formal models of the system that may be inaccurate or incomplete. As a consequence, they generally cannot establish strict correctness of the real system. The difference between the real and modeled worlds is a potential source of error that attends all uses of mathematical modeling in engineering (e.g., in numerical aerodynamics or stress calculations) and that must be controlled by validating the models concerned. Model validation, defined as the process of determining the degree to which a model is an accurate representation of the real system from the perspective of the intended uses of the model, is often required to either choose among alternative models or decide whether a model is acceptable or not before it is used for engineering analysis and design. Although there are many existing model validation methods which are mainly based on testing techniques and involve validating the trained model against a family of test datum, e.g., \textsuperscript{[17]}.\textsuperscript{[25]}, \textsuperscript{[27]}, \textsuperscript{[29]}, \textsuperscript{[33]}, \textsuperscript{[36]}, \textsuperscript{[37]}, two fundamental and pressing problems still remain to be settled when model based methods are used to verify real systems formally.

1) The first one is how to characterize the overall discrepancy between the abstraction and the real system formally.

2) The second one is what the relationship between properties inferred by the abstraction and ones from the real system is.

The solutions to these two problems are very important for the use of model-based methods to verify safety-critical systems, especially for black-box safety-critical systems. Generally, black-box systems are systems that no information is provided about what exactly makes them arrive at their predictions\textsuperscript{[34]}. Highly successful machine learning and artificial intelligence models are typical examples that often apply in a black-box manner in safety-critical systems such as medical diagnosis\textsuperscript{[15]}, \textsuperscript{[38]}. Up to present, although some works on formal verification or falsification of black-box systems exist, e.g., \textsuperscript{[35]}.\textsuperscript{[40]}, \textsuperscript{[42]}, \textsuperscript{[50]}, they mainly focus on verifying whether the black-box system with given inputs satisfies or violates user-defined properties. In contrast, the objective of our work is to estimate the set of safe inputs leading to the satisfiability of the safety specification, thereby providing users the conditions under which the black-box system works well.

Given a family of independent and identically distributed
samples extracted from the input region and their corresponding outputs, we propose a method for under-approximating the set of inputs that lead the black-box system to respect the safety specification via solving the two problems mentioned above. Akin to [16], our method falls within the PAC learning framework [22]. The computed under-approximation comes with statistical soundness in the sense that its validation is expressed using error probabilities and error confidence. It is termed as PAC under-approximation of safe inputs. With a confidence, it is a set of inputs leading to the satisfiability of the safety specification with the probability being larger than a threshold. Such an under-approximation is obtained by computing so-called PAC models of the black-box system with respect to the specified safety specification. In particular, the PAC model is computed based on the scenario approach, which is finally reduced to a linear programming problem. The linear programming is constructed based on the given family of input samples and their corresponding outputs. After obtaining such a model, we infer an under-approximation of the set of inputs, which drive the modeled system to respect the safety specification. Then, we compute the probability of safe inputs in this under-approximation. If the probability is larger than a threshold, this under-approximation is a PAC under-approximation of safe inputs. Some case studies demonstrate the merits of our approach.

The contributions of this work are summarized as follows:

1) We propose a linear programming based method for learning a model from a given family of independent and identically distributed samples extracted from the input region and their corresponding outputs. The learned model is termed a PAC model of the black-box system with respect to the safety specification. Its validation does not rely on external validation datasets. It is characterized by two informative measures: confidence level $\beta \in (0, 1)$ and error level $\epsilon \in (0, 1)$. This is the major contribution of the present work.

2) We leverage these two informative measures i.e., $\beta$ and $\epsilon$, to characterize the discrepancy between the under-approximation inferred by the PAC model and the one for the black-box system. If the discrepancy, which is characterized in terms of probability, is smaller than a threshold, the under-approximation inferred by the model is a PAC under-approximation of safe inputs for the black-box system.

3) Some case studies are employed to demonstrate the usefulness of our method.

A closely related work to the present work is [32], which proposed a data-driven procedure based on Gaussian Process Regression Models for efficient simulation-based, statistical verification without the reliance upon exhaustive simulations. It is well-known that one limitation with Gaussian process regression is the Gaussian assumption of the process itself. There are many types of data that are not well modelled under this assumption. Our method in this paper relies on scenario approaches for constructing PAC models to estimate the set of safe inputs, and is not limited to any particular distribution.

Another related works are the ones on searching for counterexamples, e.g., [4], [12], [39], [42]. Rather than identifying the set of operating configurations for which the system will satisfy the safety specification, these falsification methods utilize optimization techniques to find counterexamples such that the system fails to meet the safety specification. However, falsification methods may have to solve non-convex optimization. Just as optimization may repeatedly fall into the same local optimums, falsification searches may inadvertently return the same counterexample, even if they are initialized at different starting configurations. While these methods are extremely useful for quickly finding a single counterexample, they may be not perfectly suited to problems where multiple counterexamples exist and the goal is to identify the entire set of unsatisfactory operating conditions.

The structure of this paper is presented as follows. Section III formulates the black-box system and the problem of interest in this paper. Section IV introduces the scenario optimization and illustrates our scenario optimization-based approach for computing PAC models and PAC under-approximations of safe inputs. After showing the evaluation results of our approach in Section V we conclude this paper in Section VI.

II. Preliminaries

In this section, we introduce the black-box system as well as the concepts of PAC model and PAC under-approximation of safe inputs for the black-box system.

In science, computing, and engineering, a black-box system is a device, system or object that can be viewed in terms of its inputs and outputs, whose implementation is opaque, without any knowledge of its internal working mechanism. Almost anything might be referred to as a black-box system: a transistor, an algorithm, or the human brain. A black-box system is often postulated to act on a vector of input variables, $x \in \mathbb{R}^n$, to produce an output, $y \in \mathbb{R}^m$. Mathematically, it can be characterized by

$$y = B(x),$$

where $x$ is the input and $y$ is the output of the black-box system, and $B$ is the black-box mapping from $x$ to $y$. However, we have no knowledge of the mapping $B(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$.

All we know about the black-box system of interest in this paper is a family of datum, which is formally formulated in Assumption 1.

Assumption 1: Suppose now that $X_0$ is endowed with a $\sigma$-algebra $\mathcal{D}$ and that a probability $\Pr$ over $\mathcal{D}$ is assigned. All we know about the black-box system (1) is a family of independent and identically distributed inputs $(x_i)_{i=1}^N$ extracted from the input region $X_0$ according to the probability distribution $\Pr$ and their corresponding outputs $(y_i)_{i=1}^N$. Mathematically, $y_i = B(x_i), i = 1, \ldots, N$.

The safety of the black-box system (1) of interest in this paper is formalized in Definition 1 as follows:
Definition 1: The black-box system $\mathbb{I}$ is safe with an input $x_0 \in X_0$, if its output falls within the safe region $S$, i.e.,

$$B(x_0) \in S,$$

where $X_0 \subseteq \mathbb{R}^n$, $S = \{y \in \mathbb{R}^m \mid \phi(y) \leq 0\}$ and $\phi(\cdot)$ is a mapping from $\mathbb{R}^m$ to $\mathbb{R}$.

Given an input $x_0 \in X_0$, how can we verify whether this input is safe for the black-box system $\mathbb{I}$ without actually running this system? One solution is to compute the set of inputs rendering the black-box system $\mathbb{I}$ safe. If the input $x_0$ belongs to this set, the black-box system $\mathbb{I}$ is safe with the input $x_0$. For this sake, we often resort to mathematical models. A desirable mathematical model for the black-box system $\mathbb{I}$ is an approximate model of the black-box system $\mathbb{I}$ with respect to the safe region $S$.

In Definition 2, $\varepsilon$ can be constant or input dependent. If we obtain a function $f$ in Definition 2, we can identify the set $\{x \in X_0 \mid f(x) + \varepsilon \leq 0\}$ as an under-approximation of the set of inputs rendering the black-box system $\mathbb{I}$ safe. However, it is quite challenging, and often impossible to compute such an approximate model due to the absence of internals of the black-box system $\mathbb{I}$. Since a black-box system of interest in this paper refers to a system for which we can only observe its inputs and outputs, based on the given family of independent and identically distributed samples extracted from the input region $X_0$ and their corresponding outputs in Assumption 1, we resort to the computation of PAC models of the black-box system $\mathbb{I}$ and PAC under-approximations of safe inputs. The PAC model falls within the framework of PAC learning [22] and is formally defined below.

Definition 3: A function $f : \mathbb{R}^n \to \mathbb{R}$ is a PAC model of the black-box system $\mathbb{I}$ with respect to the safe region $S$, $\varepsilon > 0$, $\epsilon \in (0, 1)$ and $\beta \in (0, 1)$ (i.e., $\text{PACM}(S, \varepsilon, \epsilon, \beta)$), if with confidence no smaller than $1 - \beta$,

$$\text{Pr}(C) \geq 1 - \epsilon,$$

where $C = \{x \in X_0 \mid |f(x) - \phi(B(x))| \leq \varepsilon\}$.

Based on the PAC model in Definition 3, we define PAC under-approximations of the set of inputs leading the black-box system $\mathbb{I}$ to respect the safety specification $S$.

Definition 4: A set $S \subseteq X_0$ of inputs is called a PAC under-approximation of safe inputs for the black-box system $\mathbb{I}$ with respect to the safe region $S$, $\varepsilon' \in (0, 1)$ and $\beta \in (0, 1)$ (i.e., $\text{PACUAS}(S, \varepsilon', \beta)$), if with confidence of at least $1 - \beta$,

$$\text{Pr}'(S') \geq 1 - \varepsilon',$$

where $\text{Pr}' = \frac{\text{Pr}'}{\text{Pr}(S)}$ is the probability distribution defined on the set $S'$, and $S' \subseteq S$ is a set of inputs which the black-box system $\mathbb{I}$ is safe with respect to.

Similarly, we present the concept of PAC under-approximation of the set of inputs leading to the violation of the safety specification $S$ for the black-box system $\mathbb{I}$.

Definition 5: A set $U \subseteq X_0$ of inputs is called a PAC under-approximation of unsafe inputs for the black-box system $\mathbb{I}$ with respect to the safe region $S$, $\varepsilon' \in (0, 1)$ and $\beta \in (0, 1)$ (i.e., $\text{PACUAU}(S, \varepsilon', \beta)$), if with confidence of at least $1 - \beta$,

$$\text{Pr}'(U') \geq 1 - \varepsilon',$$

where $\text{Pr}' = \frac{\text{Pr}'}{\text{Pr}(U)}$ is the probability distribution defined on the set $U'$, and $U' \subseteq U$ is a set of inputs driving the black-box system $\mathbb{I}$ to fall outside the safe region $S$.

Although the computation of $\text{PACUAS}(S, \varepsilon', \beta)$ is not the focus of this paper, its computation is instrumental in detecting counterexamples, e.g., [12].

If $\beta$ in Definition 5 is extremely small (e.g., smaller than $10^{-20}$), then we have a priori practical certainty that the probability of inputs such that the PAC model $f$ is an approximate model of the black-box system $\mathbb{I}$ with respect to $\varepsilon > 0$ and the safe region $S$ is larger than $1 - \epsilon$. This observation is applicable to PAC under-approximations in Definition 4 and 5 as well. In Definition 3, 4 and 5 the probability $\beta$, which is related to $\text{Pr}$, refers to the confidence associated to the randomized solution algorithm.

For all experiments in this paper, we use the uniform distribution $\text{Pr}$ on $X_0$ to illustrate our method, i.e., the family of inputs $(x_i)_{i=1}^N$ is extracted according to the uniform distribution, although our method is not confined to this particular distribution (This distinguishing feature is reflected in Remark 3 in Subsection III-A). To some extent, the use of uniform distribution on $X_0$ is reasonable in our framework. Every state in $X_0$ is of equal importance especially for safety-critical systems, since any state in $X_0$ leading to the violation of the safety specification will result in a system failure.

III. PACUAS Generation

Based on a family of independent and identically distributed input samples and their corresponding outputs in Assumption 1 in this section we present our method for computing PAC under-approximations of safe inputs. The computational method is based on the scenario approach yielding a linear program to be solved for computing the required PAC models.

A. Scenario Optimization

We start by providing a brief introduction to the scenario optimization, which is a technique for computing solutions to robust optimization problems based on finite randomization of the constraints. Concretely, consider the following robust optimization problem:

$$\min_{\gamma \in \mathbb{R}^m} \mathbb{E}[\gamma]$$

s.t. $f_\delta(\gamma) \leq 0, \forall \delta \in \Delta$.

where $f_\delta(\gamma)$ are convex functions over the $m$-dimensional optimization variable $\gamma$ for every $\delta$ in the set $\Delta$. The scenario optimization approach for solving (2) goes as follows [6].

Definition 6 (Scenario optimization): Extract $N$ independent and identically distributed samples $\delta^{(1)}, \ldots, \delta^{(N)}$ from $\Delta$.
according to the probability distribution P and solve the convex program (3):
\[
\min_{\gamma \in \mathbb{R}^m} c^T \gamma \\
\text{s.t. for each } i = 1, \ldots, N : \\
f_{\delta^{(i)}}(\gamma) \leq 0.
\]
(3) is a relaxation of (2) and solving (3) to obtain an approximate solution to (2) is called scenario optimization of (2). Though (3) relaxes (2) in that it only considers a finite subset of the infinitely many convex constraints constituting (2), a mathematically rigorous relation between the solutions of the two systems can be drawn. The following theorem shows that the solution \( \gamma_i^{\ast} \) to (3) is bound to satisfy all constraints in (2) except a user-chosen fraction that tends to zero as \( N \) increases.

**Theorem 1:** \([3]\) Select an error level \( \varepsilon \in (0,1) \) and a confidence level \( \beta \in (0,1) \). If (3) is feasible and attains a unique optimal solution, and
\[
\varepsilon \geq \frac{2}{N} (\ln \frac{1}{\beta} + m),
\]
where \( m \) is the number of variables \( \gamma \), then with at least \( 1 - \beta \) confidence, \( \gamma_i^{\ast} \) satisfies all constraints in \( \Delta \) but at most a fraction of probability measure \( \varepsilon \), i.e., \( P(\{ \delta \in \Delta \mid f_{\delta}(\gamma) \geq 0 \}) \leq \varepsilon \).

In Theorem 1, \( 1 - \beta \) is the \( N \)-fold probability \( P^N \) in \( \Delta^N = \Delta \times \Delta \times \ldots \times \Delta \), which is the set to which the extracted sample \( (\delta^{(1)}, \ldots, \delta^{(N)}) \) belongs.

**Remark 1:** Since a unique optimal solution can be selected according to Tie-break rule if multiple optimal solutions occur for (3), Theorem 1 still holds if the uniqueness of optimal solutions to (3) in Theorem 1 is removed (6).

**Remark 2:** Absolute confidence, i.e. \( \beta = 0 \), requires \( N = \infty \). However, \( \beta \) appears under the sign of logarithm in (4) and can consequently be made very small, like \( 10^{-10} \) or \( 10^{-20} \), without increasing \( N \) significantly. Moreover, we observe from Theorem 1 that the number \( N \) of required samples does not depend on the dimension of the variable \( \delta \). This facilitates application of the scenario approach to systems with high dimensional universally quantified variable \( \delta \).

**Remark 3:** The bound (4) is probability independent, i.e., it holds irrespective of the underlying probability P, and can therefore be applied even when P is unknown.

**B. PACM Generation**

Based on the scenario optimization introduced in Subsection III-A, we now elucidate our approach to compute PAC models of the black-box system (1).

We first select an appropriate model template \( f(c_1, \ldots, c_l, x) \) such that \( f(c_1, \ldots, c_l, x) \) is for every \( x \in \mathbb{R}^n \) a linear function in \( c_1, \ldots, c_l \), where \( (c_i)_{i=1,\ldots,l} \) are unknown parameters and \( l \geq 1 \) is a positive integer. For ease of exposition, we use \( c \) to denote \( (c_1, \ldots, c_l) \) in the remainder of this paper. The selected model templates can be either polynomial functions or non-polynomial functions such as radial basis functions (30). For black-box systems, whose internal working mechanism is described by differential equations, these templates are appropriate since Taylor models of either the polynomial or non-polynomial form are often used to approximate reachable sets, e.g., (1). (9). (21). (28). (44). Note that how to select the template intelligently is not the focus of this paper.

Ideally, we wish to find \( c \) such that the discrepancy, which is characterized by a value \( \varepsilon \), between \( f(c, x) \) and \( \phi(B(x)) \) is as small as possible over \( x \in X_0 \). The search for such \( c \) can be formulated as the following robust optimization:
\[
\min_{c, \varepsilon} \varepsilon \\
\text{s.t. } f(c, x) - \phi(B(x)) \leq \varepsilon, \forall x \in X_0, \\
- U \leq c_i \leq U, i = 1, \ldots, l, \\
0 \leq \varepsilon \leq U_\varepsilon,
\]
where \( U \in \mathbb{R}_{\geq 0} \) is a pre-specified upper bound for \( c_i, i = 1, \ldots, l \), and \( U_\varepsilon \in \mathbb{R}_{\geq 0} \) is a pre-specified upper bound for \( \varepsilon \).

It is well known that the robust optimization (5) is very challenging to solve. What is even worse, \( B(x) \) is often unknown. Therefore, in this paper we resort to scenario approaches introduced in Subsection III-A for solving the robust optimization (5) from the statistical perspective.

In the framework of scenario approaches, based on the given family of input samples \( (x_i)_{i=1}^N \) and their corresponding outputs \( (y_i)_{i=1}^N \) in Assumption 1 we obtain the following linear program:
\[
\min_{c, \varepsilon} \varepsilon \\
\text{s.t. for each } i = 1, \ldots, N : \\
\phi(B(x_i)) - f(c, x_i) \leq \varepsilon, \\
f(c, x_i) - \phi(B(x_i)) \leq \varepsilon, \\
- U \leq c_i \leq U, i = 1, \ldots, l, \\
0 \leq \varepsilon \leq U_\varepsilon,
\]
which is equivalent to
\[
\min_{c, \varepsilon} \varepsilon \\
\text{s.t. for each } i = 1, \ldots, N : \\
\phi(y_i) - f(c, x_i) \leq \varepsilon, \\
f(c, x_i) - \phi(y_i) \leq \varepsilon, \\
- U \leq c_i \leq U, i = 1, \ldots, l, \\
0 \leq \varepsilon \leq U_\varepsilon,
\]
where \( U \in \mathbb{R}_{\geq 0} \) is a pre-specified upper bound for \( c_i, i = 1, \ldots, l \), and \( U_\varepsilon \in \mathbb{R}_{\geq 0} \) is a pre-specified upper bound for \( \varepsilon \).

Denote the optimal solution to (7) by \( (c^\ast, \varepsilon^\ast) \).

If \( \varepsilon \geq \frac{2}{N} (\ln \frac{1}{\beta} + l + 1) \), \( f(c^\ast, x) \) is a PAC model of the black-box system (1) with respect to the safe region \( S, \varepsilon^\ast, \epsilon \) and \( \beta \). This is formally stated in Theorem 2.

**Theorem 2:** Let \( (c^\ast, \varepsilon^\ast) \) be an optimal solution to (7) and
\[
N \geq \frac{2}{\epsilon} (\ln \frac{1}{\beta} + l + 1) .
\]
Then, \( f(e^*, x) \) is a PACM(\( S, \varepsilon^*, \epsilon, \beta \)).

**Proof 1:** Let \((e^*, \varepsilon^*')\) be the unique solution to (7). According to Theorem 1, we have that with confidence of at least \(1 - \beta\),

\[
\Pr\{ (x \in X_0 \mid -\varepsilon^* \leq \phi(B(x))) - f(x) \leq \varepsilon^* \} \geq 1 - \epsilon.
\]

Therefore, according to Definition 3, \( f(e^*, x) \) is a PACM(\( S, \varepsilon^*, \epsilon, \beta \)). When the optimal solution to (7) is not unique, the Tie-Break Rule suggested in [6] can help attain a unique solution. We can choose \( \varepsilon := \varepsilon^* \) and then solve (7) subject to the only variable \( \varepsilon \). Obviously, the optimal value of \( \varepsilon \) is still \( \varepsilon^* \) and is unique, and the conclusion still holds.

**C. PACUA Generation**

After computing a PAC model \( f(e^*, x) \) of the black-box system (1) with respect to the safe region \( S, \varepsilon^*, \epsilon \) and \( \beta \), we in this subsection introduce our method for computing PAC under-approximations of safe inputs.

As mentioned before, if \( f(e^*, x) \) satisfies that

\[
-\varepsilon^* \leq \phi(B(x)) - f(e^*, x) \leq \varepsilon^*
\]

for \( x \in X_0 \), the set

\[
\{ x \in X_0 \mid f(e^*, x) + \varepsilon^* \leq 0 \}
\]

is a set of safe inputs such that their corresponding outputs fall within the safe region \( S \). In contrast, the set

\[
\{ x \in X_0 \mid f(e^*, x) - \varepsilon^* > 0 \}
\]

is an under-approximation of the set of inputs such that their corresponding outputs fall outside the safe region \( S \). However, the PAC model \( f(e^*, x) \) only tells us that with confidence of at least \(1 - \beta\),

\[
\Pr(C) \geq 1 - \epsilon,
\]

where \( C = \{ x \in X_0 \mid | f(e^*, x) - \phi(B(x))| \leq \varepsilon^* \} \). The characterization of the robustness of the under-approximation \( \{ x \in X_0 \mid f(e^*, x) + \varepsilon^* \leq 0 \} \) is presented in Lemma 1.

**Lemma 1:** Let \( \Pr(S) = \epsilon^* \), where \( S = \{ x \in X_0 \mid f(e^*, x) + \varepsilon^* \leq 0 \} \). Then with confidence of at least \(1 - \beta\),

\[
\Pr(S') \geq \epsilon^* - \epsilon,
\]

where \( S' = \{ x \in S \mid \phi(B(x)) \leq 0 \} \) is the set of safe inputs in \( S \) leading the black-box system (1) to respect the safety specification \( S \).

**Proof 2:** Since with confidence of at least \(1 - \beta\),

\[
\Pr(C) \geq 1 - \epsilon,
\]

where \( C = \{ x \in X_0 \mid | f(e^*, x) - \phi(B(x))| \leq \varepsilon^* \} \), we have that \( \Pr\{ (x \in X_0 \mid \phi(B(x)) > f(e^*, x) + \varepsilon^*) \} \leq \epsilon \), with confidence of at least \(1 - \beta\). Therefore, we have that with confidence of at least \(1 - \beta\),

\[
\Pr(S') \geq \epsilon^* - \epsilon,
\]

where \( S' = \{ x \in S \mid \phi(B(x)) \leq 0 \} \).

Consequently, we infer that \( \{ x \in X_0 \mid f(e^*, x) + \varepsilon^* \leq 0 \} \) is a PAC under-approximation of safe inputs for the black-box system (1) with respect to the safe region \( S, \varepsilon^*, \beta \).

**Theorem 3:** Let \( \Pr(S) = \epsilon^* \), where \( S = \{ x \in X_0 \mid f(e^*, x) + \varepsilon^* \leq 0 \} \). The set \( S \) is a PACUAS(\( S, \varepsilon^*, \beta \)).

**Proof 3:** The conclusion is an immediate result of Lemma 1 and Definition 4.

According to Theorem 3, the smaller \( \frac{\varepsilon}{\epsilon^*} \) is, the larger the probability of inputs in \( \{ x \in X_0 \mid f(e^*, x) + \varepsilon^* \leq 0 \} \) leading to the satisfiability of the safe region \( S \).

Similarly, we have the following lemma for the set \( \{ x \in X_0 \mid f(e^*, x) - \varepsilon^* > 0 \} \), which possibly is a set of inputs with their outputs falling outside the safe region \( S \).

**Lemma 2:** Let \( \Pr(U) = \epsilon^* \), where \( U = \{ x \in X_0 \mid f(e^*, x) - \varepsilon^* > 0 \} \). Then with confidence of at least \(1 - \beta\),

\[
\Pr(U') \geq \epsilon^* - \epsilon,
\]

where \( U' = \{ x \in U \mid \phi(B(x)) > 0 \} \) is the set of unsafe inputs in \( S \) leading the black-box system (1) to violate the safety specification \( S \).

According to Lemma 2 and Definition 5, we state that the set \( \{ x \in X_0 \mid f(e^*, x) - \varepsilon^* > 0 \} \) is a PACUAU(\( S, \varepsilon^*, \beta \)).

**Theorem 4:** Let \( \Pr(U) = \epsilon^* \), where \( U = \{ x \in X_0 \mid f(e^*, x) - \varepsilon^* > 0 \} \). The set \( U \) is a PACUAU(\( S, \varepsilon^*, \beta \)).

The smaller \( \frac{\varepsilon}{\epsilon^*} \) is, the larger the probability of inputs in the set \( \{ x \in X_0 \mid f(e^*, x) - \varepsilon^* > 0 \} \) with their outputs falling outside the safe region \( S \).

The procedure of computing PAC under-approximations of safe inputs for the black-box system (1) is formally summarized in Algorithm 1.

**Algorithm 1** Computing PAC under-approximations of safe inputs for black-box systems

**Require:** the black-box system (1); \( (x_i, y_i)_{i=1}^N \); a family of independent and identically distributed inputs \( (x_i)_{i=1}^N \) and their corresponding outputs \( (y_i)_{i=1}^N \); \( y \in \mathbb{R}^m \mid \phi(y) \leq 0 \); the safe region; \( \beta \in (0, 1) \); confidence measure; \( \epsilon' \in (0, 1) \); the probability measure for characterizing the computed PAC under-approximation.

**Ensure:** PACUAS\( (S, \varepsilon', \beta) \).

1) Select a PACM template \( f(e, x) \);
2) if an optimal solution \( (e^*, \varepsilon^*) \) is computed via solving (7) then
   Compute the error level \( \epsilon \) according to (8);
   PACM\( (S, \varepsilon^*, \epsilon, \beta) := f(e^*, x) \);
   \( \epsilon' := \Pr\{ (x \in X_0 \mid f(e^*, x) + \varepsilon^* \leq 0) \}; 
   \) if \( 1 - \frac{\epsilon}{\epsilon'} \geq 1 - 1 - \beta \) then
   PACUAS\( (S, \varepsilon', \beta) := \{ x \in X_0 \mid f(e^*, x) + \varepsilon^* \leq 0 \}\); (Theorem 3);
   Return PACUAS\( (S, \varepsilon', \beta) \);
3) else
   Return ‘Change the PACM template and try another round’;
4) end if
Remark 4: Given a PAC model template, if Alg. 1 failed to generate a PACUAS($S, \epsilon^*, \beta$), one can try another template (e.g., if a template of the polynomial form in $x$ is employed, we can change the currently used template to a polynomial template of higher degree.) with the given $N$ datum to perform computations. This will reduce the value $\epsilon$ and consequently reduce the conservativeness in modeling the black-box system \cite{1} with respect to the safe region $S$. If possible, we can also extract more samples from the input region $X_0$ for performing computations to improve the error level $\epsilon$ of the PAC model. Another solution is to drop some bad samples from the family of $N$ input samples and their corresponding outputs for improving the value $\epsilon$ as in \cite{3, 7}, which extend scenario approaches to solving the chance-constrained optimization.

We use a simple example to enhance the understanding of Alg. 1.

Example 1: The black-box system in this example is illustrated in Fig. 1. The black-box system is controlled by a two-dimensional ordinary differential equation:

\[
\begin{align*}
\frac{dx_1}{dt} &= -x_1, \\
\frac{dx_2}{dt} &= -x_2,
\end{align*}
\]

where $X_0 = [-0.5, -0.4] \times [-0.5, -0.4]$. The system is safe with respect to the input $x_0 \in X_0$ if the state $y = (x_1(1), x_2(1))^T$ at time $t = 1$ belongs to the safe set $S = \{ y \in \mathbb{R}^2 \mid (y_1 + 0.16)^2 + (y_2 + 0.16)^2 - 0.002 \leq 0 \}$. Note that the explicit presentation of the ordinary differential equation (9) is only for obtaining outputs of the black-box system and making comparisons. We do not use it to compute under-approximations of the set of safe inputs.

Let the number of datum $(x_i, y_i)_{i=1}^N$ be 10611 and the confidence level $\beta = 10^{-20}$. Both the input samples $(x_i)_{i=1}^{10611}$ and their corresponding outputs $(y_i)_{i=1}^{10611}$ are visualized in Fig. 2. We further assume that the PAC model template is $f(c, x) = c_1 + c_2 x_1 + c_3 x_2 + c_4 x_1^2 + c_5 x_1 x_2 + c_6 x_2^2$, where $c = (c_1, c_2, c_3, c_4, c_5, c_6)$. Therefore, the resulting linear program is

\[
\begin{align*}
\min_{\epsilon, c} & \quad \epsilon \\
\text{s.t.} & \quad (y_i) - f(c, x_i) \leq \epsilon \\
& \quad (c, x_i) - f(c, x_i) \leq \epsilon, \\
& \quad 10 \leq c_j \leq 10, j = 1, \ldots, 6, \\
& \quad 0 \leq \epsilon \leq 10.
\end{align*}
\]

In this example $U = U_0 = 10$. The number of variables in the linear optimization (10) is 7.

Via solving (10), we obtain that

\[
f(x) = 0.0492 + 0.1177 x_1 + 0.1177 x_2 + 0.1353 x_1^2 + 0.1353 x_2^2
\]

and $\epsilon^* = 1.55 \times 10^{-8}$. According to (8) in Theorem 2 we have $\epsilon = 0.01$. Therefore, $f(x)$ is a PACM($S, \epsilon^*, \beta$). Its level sets are illustrated in Fig. 3. The error between $f(x)$ and $\phi(B(x)) = (\frac{x}{\epsilon} + 0.16)^2 + (\frac{x}{\epsilon} + 0.16)^2 - 0.002$ over $X_0$, i.e. $|f(x) - \phi(B(x))|$ for $x \in X_0$, is also visualized in Fig. 3.

The visualized result in Fig. 3 shows that $f(x)$ approximates $\phi(B(x))$ very well over $X_0$.

An under-approximation of safe inputs induced by the PAC model $f(x)$ is

\[
S = \{ x \in X_0 \mid f(x) + \epsilon^* \leq 0 \}.
\]

The set $\{ x \in \mathbb{R}^2 \mid f(x) + \epsilon^* \leq 0 \}$ is illustrated in Fig. 2. It is observed from Fig. 2 that $\{ x \in X_0 \mid f(x) + \epsilon^* \leq 0 \} = X_0$ and therefore $\text{Pr}(S) = \epsilon^* = 1$. Since $1 - \frac{1}{2} = 0.99$, we obtain $Pr(S^c) \geq 0.99$, where $S^c = \{ x \in S \mid \phi(B(x)) \leq 0 \} \subset S$ is a set of inputs rendering the black-box system safe. Therefore, $S$ is a PACUAS($S, 0.01, 10^{-20}$) according to Theorem 3.

**IV. Experiments**

In this section we evaluate Alg. 1 on several case studies. Parameters that determine the performance of Alg. 1 are shown in Table 1. All computations were performed on an i7-7500U 2.70GHz CPU with 32G RAM running Windows 10.

Example 2: In this example we consider a black-box system illustrated in Fig. 4 whose internal working mechanism is described by a delay differential equation of the following form:

\[
\begin{align*}
\frac{dx_1}{dt} &= ax_1(t)(1 - \frac{x_1(t)}{m}) + bx_1(t)x_2(t) + cx_2(t), \\
\frac{dx_2}{dt} &= cx_2(t) + dx_1(t - \tau)x_2(t - \tau),
\end{align*}
\]

where $\tau = 0.1, a = 0.25, b = -0.01, c = -1.00$ and $d = 0.01$. The initial condition $x(t)$ over $t \in [-0.1, 0]$ for this delay differential equation is a constant vector falling within $X_0$. This delay differential equation was a model for predator-prey populations, where $x_1$ and $x_2$ are the prey and predators.
Let $X_0 = \{8, 9\} \times \{6, 8\}$ and $S = \{y \in \mathbb{R}^2 \mid \frac{(y_1-10.3)^2}{20} + (y_2 - 2.8)^2 - 0.1 \leq 0\}$. The number of samples is 12411. The samples are visualized in Fig. 5. The output of the black-box system is the state of $(11)$ at time $t = 1$. For this example, we use the polynomial of degree 4 as the PAC model template. The zero level set of the system is the state of $(11)$ at time $t = 1$. For this example, we use the polynomial of degree 4 as the PAC model template. The zero level set of the computed PACM $(S, \epsilon, \epsilon, \beta)$, i.e. $\{x \in X_0 \mid f(x) + \epsilon^* = 0\}$, is visualized in Fig. 5 as well, where $\epsilon^* = 2.65 \times 10^{-8}$ and $\epsilon = 0.01$.

There are some methods for computing the probability $Pr(S)$, i.e. $[23, 26]$, where $S = \{x \in X_0 \mid f(x) + \epsilon^* = 0\}$. Herein we use Monte Carlo Methods to estimate this probability and obtain $\epsilon^* = Pr(S) \approx 0.46$. Therefore, we obtain that $Pr'(S') \geq 1 - \frac{\epsilon^*}{\epsilon} = 1 - 0.46 \geq 0.978$, where $S' \subseteq S$ is presented in Definition 4 and is a set of inputs in $S$ rendering the black-box system safe. Therefore, $\{x \in X_0 \mid f(x) + \epsilon^* \leq 0\}$ is a PAC$(S, 0.022, 10^{-20})$.

We also extract $10^4$ independent and identically distributed inputs from the computed PACUS$(S, 0.022, 10^{-20})$ to check their outputs. Note that this family of inputs is independent of the family of inputs used above. All of their outputs fall within the safe region $S$. These inputs and outputs are illustrated in Fig. 5 as well.

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>$\dim u$</th>
<th>$\epsilon'$</th>
<th>$\epsilon$</th>
<th>$\beta$</th>
<th>$N$</th>
<th>$m$</th>
<th>$U$</th>
<th>$T_L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex. 3</td>
<td>2</td>
<td>0.05</td>
<td>0.01</td>
<td>$10^{-20}$</td>
<td>12411</td>
<td>16</td>
<td>100</td>
<td>63.15</td>
</tr>
<tr>
<td>Ex. 5</td>
<td>2</td>
<td>0.05</td>
<td>0.01</td>
<td>$10^{-20}$</td>
<td>10611</td>
<td>7</td>
<td>100</td>
<td>7.09</td>
</tr>
<tr>
<td>Ex. 6</td>
<td>2</td>
<td>0.05</td>
<td>0.01</td>
<td>$10^{-20}$</td>
<td>12411</td>
<td>16</td>
<td>100</td>
<td>64.66</td>
</tr>
<tr>
<td>Ex. 7</td>
<td>2</td>
<td>0.05</td>
<td>0.01</td>
<td>$10^{-20}$</td>
<td>15011</td>
<td>29</td>
<td>100</td>
<td>98.69</td>
</tr>
<tr>
<td>Ex. 8</td>
<td>101</td>
<td>0.05</td>
<td>0.01</td>
<td>$10^{-20}$</td>
<td>398104</td>
<td>153</td>
<td>100</td>
<td>127.92</td>
</tr>
</tbody>
</table>

Fig. 3. An illustration for PACM$(S, 1.55 \times 10^{-8}, 0.01, 10^{-20})$. Above: level sets of $f(x)$ over $X_0$. Below: $|f(x) - \phi(B(x))|$ over $X_0$.

Fig. 4. The black-box system for Example 2.

Input: $x_i = (x_i(1-0.1), x_i(1-0.1))$

Output: $y = (y_0(1), y_0(1))$

\[ \frac{dx_1}{dt} = ax_1(t) + bx_2(t) y_2(t), \]

\[ \frac{dx_2}{dt} = cx_2(t) + dx_1(t) - ty_2(t - r). \]

Fig. 5. An illustration of the computed PACUS$(S, 0.022, 10^{-20})$ for Example 2. Above: Green points denote samples $(x_i)^{2411}_{i=1}$; Red points denote outputs $(y_i)^{2411}_{i=1}$; Black curve denotes the zero-level set $(x \in X_0 \mid f(x) + \epsilon^* = 0)$; Yellow curve denotes the boundary of the safe region $S$. Below: Green points denote inputs extracted from PACUS$(S, 0.022, 10^{-20})$. Black curve denotes the boundary of PACUS$(S, 0.022, 10^{-20})$. Red points denote the outputs corresponding to the family of inputs extracted from PACUS$(S, 0.022, 10^{-20})$. Yellow curve denotes the boundary of $S$. For systems modeled by ordinary differential equations, there are some methods for computing under-approximations of the set of initial states rendering the system safe, e.g., [20, 44, 46, 47, 49]. Although there are some methods for under-approximating forward reachable sets for delay differential equations, e.g., [21, 45, 48], there are few works on computing under-approximations of the set of initial states rendering the system safe. The main difficulty in computing...
such under-approximations for delay differential equations may be due to the fact that their solutions do not have homeomorphism property generally [45], [48]. The method in this paper can be used for computing such under-approximations in the statistical sense for black-box systems, whose internal dynamics are subject to the delay phenomenon.

Example 3: Consider a black-box system, whose internal dynamics are described by an ordinary differential equation from [1]. This equation is generally used to describe a Van der Pol oscillator which is a standard example for nonlinear systems that have a limit cycle:

\[
\begin{align*}
\frac{dx_1}{dt} &= x_2 \\
\frac{dx_2}{dt} &= (1 - x_1^2)x_2 - x_1
\end{align*}
\] (12)

An under-approximation of safe inputs induced by PACM \(S, \varepsilon^*, \epsilon, \beta\), i.e. \(S = \{x \in X_0 \mid f(x) + \varepsilon^* \leq 0\}\), is equal to \(X_0\). Therefore, \(Pr(S) = \varepsilon^* = 1\) and consequently \(S\) is a PACUAS(\(S, 0.01, 10^{-20}\)).

An overly pessimistic under-approximations arise often due to the wrapping effect, which is the propagation and accumulation of over-approximation error through the iterative computation in the construction of reachable sets, especially for large time horizons. For instance, the method in [46] returns an empty under-approximation of the set of initial states leading this black-box system safe if we use the ordinary differential equation (12) to perform reachability analysis. In contrast, the method proposed in this paper is able to synthesize non-empty under-approximations of the set of safe inputs for black-box systems modelled by ordinary differential equations with large time horizons, albeit at the price of the computed under-approximation being only probably approximately correct.

Example 4: Consider a black-box system, whose internal working mechanism is described by a multi-layer neural network from [43]. It is illustrated in Fig. 8. The multi-layer neural network has two inputs, two outputs and one hidden layer consisting of five neurons. The activation function for the hidden layer is \(tanh\) function and \(purelin\) function is for the output layer. The weight matrices and bias vectors are as follows:

\[W[1] = \begin{bmatrix}
-0.9507 & -0.7680 \\
0.9707 & 0.0270 \\
-0.6876 & -0.0626 \\
0.4301 & 0.1724 \\
0.7408 & -0.7948
\end{bmatrix}, \quad \theta[1] = \begin{bmatrix}
1.1836 \\
-0.9087 \\
-0.3463 \\
0.2626 \\
-0.6768
\end{bmatrix},
\]

\[W[2] = \begin{bmatrix}
0.8280 & 0.6839 & 1.0645 & -0.0302 & 1.7372 \\
1.4436 & 0.0824 & 0.8721 & 0.1490 & -1.9154
\end{bmatrix},
\]

\[\theta[2] = \begin{bmatrix}
-1.4048 \\
-0.4827
\end{bmatrix}.
\]

Let \(X_0 = [0, 1] \times [0, 1]\) and \(S = \{y \in \mathbb{R}^2 \mid (y_1 + 3)^2 + (y_2 - 0.3)^2 - 1 \leq 0\}\). The number of \(s\), which are visualized in Fig. 9 is 12411. We use polynomials of degree 4 as the PAC model template. The zero level set of the computed PACM(\(S, \varepsilon^*, \epsilon, \beta\)), i.e. \(\{x \in X_0 \mid f(x) + \varepsilon^* = 0\}\), is visualized in Fig. 9 as well, where \(\varepsilon^* = 0.024\) and \(\epsilon = 0.01\).

We also use Monte Carlo Methods with \(10^4\) samples to estimate the probability \(Pr(S)\), where \(S = \{x \in X_0 \mid f(x) + \varepsilon^* \leq 0\}\), and obtain \(\varepsilon^* = Pr(S) \approx 0.74\). Therefore, \(S\) is a PACUAS(\(S, 0.0137, 10^{-20}\)). We also extract \(10^4\) independent and identically distributed inputs from the obtained PACUAS(\(S, 0.0137, 10^{-20}\)) to observe their outputs. All of their outputs fall within the safe region \(S\). These outputs are illustrated in Fig. 9 as well.
Suppose that the number of samples is 15011 and the polynomial of degree 6 is used as the PAC model template. We obtain a PACM$(S, \varepsilon^*, \epsilon, \beta)$, where $\varepsilon^* = 0.0015$ and $\epsilon = 0.01$. This results in a PACUAS$(S, 0.0134, 10^{-20})$.

There are some methods for safety verification of multi-layer perceptrons, e.g., [13], [14], [19], [24], [43]. These methods attempt to over-approximate the output range of a given white-box multi-layer perceptron. In contrast, the method in this paper targets black-box multi-layer perceptrons and focuses on the computation of under-approximations of the set of safe inputs.

![Graph](image)

**Example 5:** It is well known that classical formal methods such as model checking and theorem proving do not scale to industrial design problems. To demonstrate applicability of our approach to high-dimensional systems, we consider a scalable black-box system, which is controlled by an ordinary differential equation:

\[
\begin{align*}
\frac{dx_1}{dt} &= 1 + \frac{1}{l} \sum_{i=1}^{l} x_{i+1} + x_{i+2} \\
\frac{dx_2}{dt} &= x_3 \\
\frac{dx_3}{dt} &= -10 \sin x_2 - x_2 \\
\vdots \\
\frac{dx_{2l}}{dt} &= x_{2l+1} \\
\frac{dx_{2l+1}}{dt} &= -10 \sin x_{2l} - x_2
\end{align*}
\]

where $l = 50$.

Let $X_0 = [-0.01, 0.01]^{101}$ and $S = \{ y \in \mathbb{R}^{101} \mid \sum_{i=1}^{101} y_i \leq 1 \}$. The number of input samples is 10003. The output of the black-box system is the state at time $t = 0.2$. We use a nonlinear template of the form $c_0 + \sum_{i=1}^{101} c_i x_i + \sum_{i=2}^{51} c_{100+i} (-10 \sin x_{2(i-1)} - x_2)$. We obtain a PACM$(S, \varepsilon^*, \epsilon, \beta)$ $f(x)$, where $\varepsilon^* = 0.298$ and $\epsilon = 0.01$.

We also use Monte Carlo Methods to estimate $Pr(S)$, where $S = \{ x \in X_0 \mid f(x) + \varepsilon \leq 0 \}$, and obtain that $\varepsilon^* = Pr(S) \approx 1$. Therefore, $S$ is a PACUAS$(S, 0.01, 10^{-20})$.

Our method mainly relies on solving the linear programming (7). Fortunately, linear programming problems with hundreds of thousands or even millions of variables are routinely solved [18]. The size of the linear program (7) for computing PAC models does not depend on the dimension of the state space. In contrast, it heavily depends on $\epsilon, \beta$ and the number of unknown parameters in a pre-specified PAC model template. In this example the selection of PAC model template is based on the knowledge of ODE (13). In practice, engineering insight would facilitate the selection of PAC model templates. The dimensionality of this example demonstrates that our approach has great potential to open up a promising prospect for formal verification of industrial-scale systems by selecting appropriate $c, \beta$ and PAC model templates.

**V. CONCLUSION**

Given a family of independent and identically distributed samples extracted from the input region and their corresponding outputs, in this paper we proposed a linear programming based method for computing PAC under-approximations of safe inputs for black-box systems. The PAC under-approximation was computed based on constructing PAC models. Based on scenario approaches and the given family of datum, the PAC model was computed by solving a linear programming problem. In our method we provided a mathematical characterization of the discrepancy between the computed model and the black-box system with respect to the specified safety specification as well as a measurement of the discrepancy between the under-approximation of safe inputs for the black box system and the under-approximation inferred by the mathematical model. We finally evaluated our approach on some case studies.

**Acknowledgements.** Bai Xue was funded by CAS Pioneer Hundred Talents Program under grant No. Y8YC235015, NSFC under grant No. 61872341 and 61836005, Lei Ma was supported by JSPS KAKENHI Grant 19H04086 and Qdai-jump Research Program NO.01277, Meng Sun was supported by NSFC under grant No. 61772341 and 61836005, and the Guangdong Science and Technology Department (Grant no. 2018B010107004).

**REFERENCES**

