ProbStar Temporal Logic for Verifying Temporal Properties of Learning-enabled Systems

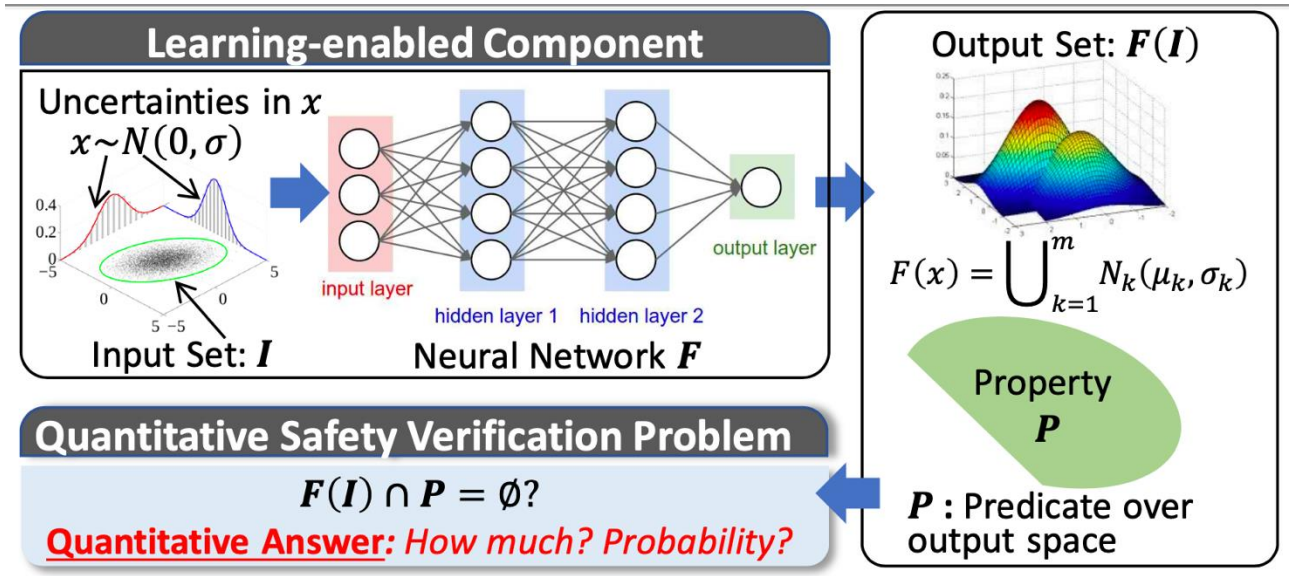
Hoang-Dung Tran

Feb 26, 2025

School of Computing

UNIVERSITY of NEBRASKA
LINCOLN

Quantitative Verification[4]



No verification approaches can precisely compute the safety probability of a network

❑ **Model a probabilistic input set using ProbStar:**

A ProbStar: $X = < c, V, P, \mathcal{N} >$

- c is the center vector
- V is the generator matrix
- $P \triangleq C\alpha \leq d$, is the predicate
- $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)^T$
- $\mathcal{N}$ is a multivariate normal distributions

❑ **Probability of a ProbStar:**

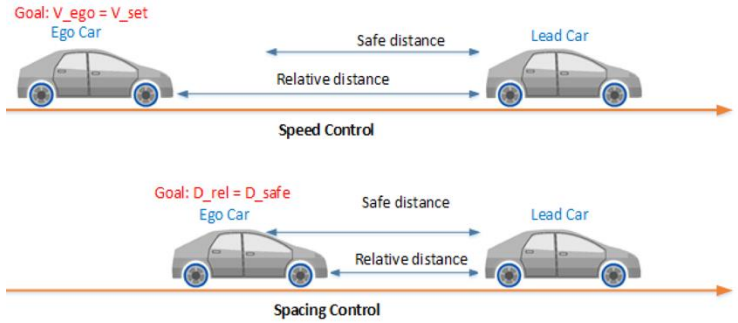
$$\Pr(X) = \Pr(C\alpha \leq d), \alpha \sim \mathcal{N}(\mu, \Sigma)$$

k-dimensional multivariate law under linear restrictions

$$f(z) = \frac{1}{l} \exp\left(-\frac{1}{2} z^T z\right) \mathrm{I}(lb \leq Cz \leq ub), z = (z_1, \dots, z_k)^T, C \in \mathbb{R}^{m \times k}, lb, ub \in \mathbb{R}^m$$

[4]Tran et al, "Quantitative Verification of Deep Neural Networks using ProbStars", HSCC'23

Verifying temporal properties[5]



Safety specification: $\boldsymbol{d = x_{lead} - x_{ego} \geq d_{safe} = 10 + 1.4 \times v_{ego}}$

Temporal Specification:

$$\varphi := \Diamond_{[0,T]}(x_{lead(t)} - x_{ego}(t) \leq d_{safe} \wedge \Box_{[0,5]}\left(\left(x_{lead}(t) - x_{ego}(t)\right) \leq d_{safe}\right))$$

"If two cars are in unsafe distance,
the unsafe condition always remains for the next 5 steps"

ProbStar Temporal Logic

- A set-based temporal logic, amenable to set-based verification

Discrete-time Syntax

I: Bounded time interval

$$\varphi := \mu_{p,q} | \neg\varphi | \varphi \wedge \varphi | \circ \varphi | \Box_I \varphi$$

Negation *Boolean* *Next time* *Always*

Semantics

$$\mathcal{R} = [X_0, X_1, \ldots, X_t],$$

$$\mathcal{C}(\mathcal{R}, t, \mu_{p,q}) = X_t \cap \mu_{p,q},\ \mathcal{C}(\mathcal{R}, t, \neg\mu_{p,q}) = X_t \cap \overline{\mu_{p,q}}$$

$$\mathcal{C}(\mathcal{R}, t, \varphi_1 \wedge \varphi_2) = \mathcal{C}(\mathcal{R}, t, \varphi_1) \wedge \mathcal{C}(\mathcal{R}, t, \varphi_2)$$

$$\mathcal{C}(\mathcal{R}, t, \varphi_1 \vee \varphi_2) = \mathcal{C}(\mathcal{R}, t, \varphi_1) \vee \mathcal{C}(\mathcal{R}, t, \varphi_2)$$

$$\mathcal{C}(\mathcal{R}, t, \circ\varphi) = \mathcal{C}(\mathcal{R}, t, +1, \varphi) \quad \mathcal{C}(\mathcal{R}, t, \Box_{[t_1,t_2]}\varphi) = \wedge_{t=t_1}^{t_2} \mathcal{C}(\mathcal{R}, t, \varphi)$$

[5]Tran et al, "ProbStar Temporal Logic for Verification of Temporal Properties of LES", HSCC'25

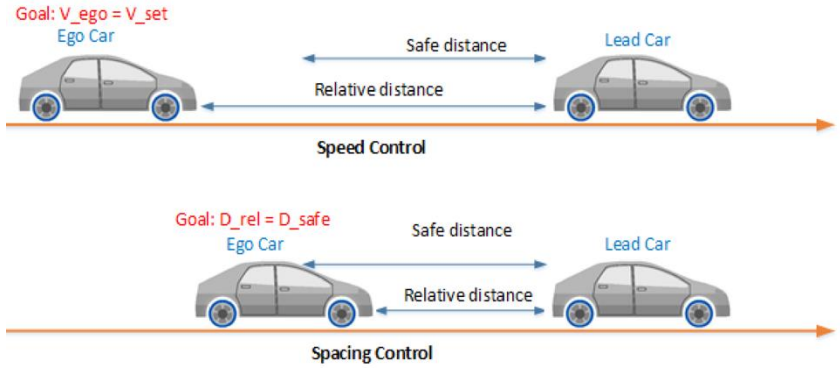
Satisfaction Probability $\mathbb{P}(\mathcal{R} \vDash \varphi)$

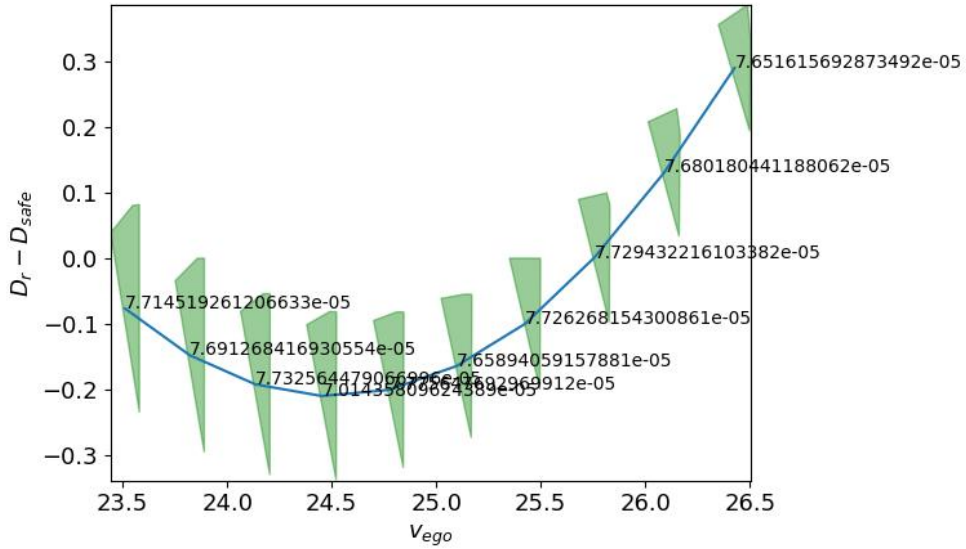
- Can be computed exactly or estimated

$$\mathcal{C}(\mathcal{R}, t, \varphi) = \bigvee_i \left(\bigwedge_j \mu_j \right) = \bigvee_i S_i \longrightarrow \text{DNF: } S_i \text{ is a ProbStar}$$

$$\mathbb{P}(\mathcal{R} \vDash \varphi) = \mathbb{P}\big(\mathcal{C}(\mathcal{R}, t, \varphi)\big) = \mathbb{P}\left(\bigvee_{i=1}^{n} S_i\right)$$
$$= \sum_{i=1}^{n} \mathbb{P}(S_i) + (-1)^1 \sum_{j \neq i}^{n} \mathbb{P}(S_i \wedge S_j)$$
$$+ (-1)^2 \sum_{i \neq j \neq k}^{n} \mathbb{P}\big(S_i \wedge S_j \wedge S_k\big) + \cdots$$
$$+ (-1)^{n-1} \mathbb{P}(S_1 \wedge \ldots \wedge S_n) \geq \max\big(\mathbb{P}(S_1), \ldots, \mathbb{P}(S_n)\big)$$

Verification Example: Le-ACC





A SAT Trace

Temporal Specification:

$$\varphi := \Diamond_{[0,T]}(x_{lead(t)} - x_{ego}(t) \leq d_{safe} \wedge \Box_{[0,5]}\left(\left(x_{lead}(t) - x_{ego}(t)\right) \leq d_{safe}\right))$$

"If two cars are in unsafe distance, the unsafe condition always remains for the next 5 steps"

T	10	20	30
$\mathbb{P}(ACC \vDash \varphi \mid X_0)$	[3.37e-09, 3.41e-09]	[0.01902, 0.01907]	[0.0121, 0.0125]