MACHINE LEARNING MEETS FORMAL VERIFICATION

Luca Bortolussi$^{1,2}$

$^1$Dipartimento di Matematica e Geoscienze
Università degli studi di Trieste, Italy
$^2$Modelling and Simulation Group
Saarland University, Saarbücken, Germany

GANDALF, September 14, Catania, Italy
COMPLEX SYSTEMS

Thex

A SIMPLE STORY (2)         Predicting the H1N1 pandemic

Network Science: Introduction

2012

Chicago
New York
Los Angeles
Houston
Toronto
Vancouver
Calgary
Indianapolis
La Glorie
San Pablo
Mexico City
Rio De Janeiro
San Juan
Bogotá
Johannesburg
Carls
Cape Town
Nairobi

Dec 30 2009

Paris
Frankfurt
Amsterdam
Rome
Munich
São Paulo
Dublin
Hong Kong
Tokyo
Naha
Bangkok
Singapore
Beijing
Manila
Sydney
Perth
Auckland

C

SmoothingMC

Robust Design

Conclusions
MODELING COMPLEX SYSTEMS

MODEL

DATA

PREDICTION by VERIFICATION
Modelling Complex Systems

Model

Uncertainty

Prediction by Verification

Data
Modelling Complex Systems

- Model
- Formal methods
- Uncertainty
- Prediction by Verification

- Data
- Machine learning

No Introduction, Background, SmoothMC, Robust Design, Conclusions
THE “PHILOSOPHICAL” POINT

**Formal Methods**

Quantitative Formal methods provide (scalable) tools to deal with both local (e.g. Stochastic Process Algebras) and global aspects (e.g. temporal logic), which are seamlessly integrated (through stochastic verification).

**Machine Learning**

Machine Learning provides us with sophisticated statistical tools to manage and refine uncertainty, in a computationally efficient way.

**Mantra**

Integrate machine learning with quantitative formal methods to produce verification tools dealing consistently with uncertainty.
THE PROBLEMS TACKLED (SO FAR)

- **Satisfaction under Uncertainty**: Compute satisfaction (probabilities) of formal properties for uncertain models.
- **Synthesis and Design**: Find model parameters satisfying (as robustly as possible) a set of formal specifications.
- **Parameter Estimation**: Learn model parameters from qualitative observations.
- **Model simplification**: Learn simplified models and correction maps.
- **Control**: Learn optimal control policies in MDPs.


OUTLINE OF THE TALK

- Quick background on stochastic modelling and formal methods
- Gaussian Processes and Smoothed Model Checking
- (Robust System Design)
A simple example: epidemic spreading

- We consider a simple model of epidemic spreading in a **homogeneous** network (with average degree \( \langle k \rangle \)).
- Each individual can be in three states: **susceptible** (S), **infected** (I), and **recovered** (R).
- The state of an agent can change due to three events:
  1. (infection) \( I + S \rightarrow I + I \) at rate \( k_i \langle k \rangle S \)
  2. (ext infection) \( S \rightarrow I \) at rate \( k_e S \)
  3. (recovery) \( I \rightarrow R \) at rate \( k_r I \)
The stochastic process behind: CTMC

We consider population CTMC models, that describe the dynamics of a population of interacting agents.

State Space
The state space is described by a set of $n$ variables $X = X_1, \ldots, X_n \in \mathbb{N}$, each counting the number of agents (jobs, molecules, ...) of a given kind.

Transitions
The dynamics is given by a set of rules of the form

$$s_1 X_1 + \ldots + s_n X_n \rightarrow r_1 X_1 + \ldots + r_n X_n,$$

with rate given by a function $f(X, \theta)$, depending on the system variables and on a set of parameters $\theta$. 
THE STOCHASTIC PROCESS BEHIND: CTMC

Rules

\[ s_1 X_1 + \ldots + s_n X_n \rightarrow r_1 X_1 + \ldots + r_n X_n \]
Update vector: \( \mathbf{v} = (r_1 - s_1, \ldots, r_n - s_n) \)
Rate: \( f(\mathbf{X}, \theta) \).

Master Equation

\[
\frac{dP(\mathbf{X}, t)}{dt} = \sum_j P(\mathbf{X} - \mathbf{v}_j, t)f(\mathbf{X} - \mathbf{v}_j, \theta)dt - \sum_j P(\mathbf{X}, t)f(\mathbf{X}, \theta)dt
\]

Simulation

We can simulate such model with standard algorithms, the most known is Gillespie’s one.
We are often interested in qualitative features of the system, such as:

1. The epidemics stopped between days 60 and 90.
2. The final fraction of infected people is between 80% and 90%.
3. The fraction of infected people remained below 45%.

We will formalise them with metric interval temporal logic.
Metric Interval Temporal Logic

- We express qualitative properties by Metric (interval) Temporal Logic.
- Linear continuous time: in experiments we observe single realisations.
- Metric bounds: we can observe a system only for a finite amount of time.

**Syntax**

\[ \varphi ::= \top \mid \mu \mid \neg \varphi \mid \varphi_1 \land \varphi_2 \mid \varphi_1 U_{[T_1,T_2]} \varphi_2, \]

As customary: \( F_{[T_1,T_2]} \varphi \equiv \top U_{[T_1,T_2]} \varphi \), \( G_{[T_1,T_2]} \varphi \equiv \neg F_{[T_1,T_2]} \neg \varphi \).

- \( F_{[T_1,T_2]} \varphi \): \( \varphi \) is **eventually** true between time \( T_1 \) and \( T_2 \).
- \( G_{[T_1,T_2]} \varphi \): \( \varphi \) is **always** true between time \( T_1 \) and \( T_2 \).
- \( \varphi_1 U_{[T_1,T_2]} \varphi_2 \): \( \varphi_1 \) is true **until** \( \varphi_2 \) becomes true between time \( T_1 \) and \( T_2 \).
MITL FOR EPIDEMIC SPREADING

\[ G_{[0,200]}(X_I < 45) \]
the fraction of infected never exceeds 45% in the first 200 time units. This upper bounds the number of active infecting individuals.

\[ F_{[22,40]}(X_I > 35) \]
between time 22 and 40, the fraction of infected exceeds 35%. This locates the infection peak between time 22 and 40.

\[ (X_I > 0)U_{[100,120]}(X_I = 0) \]
the epidemic stops between time 100 and 120.

\[ G_{[90,200]}(82 < X_R < 88) \]
the final fraction of recovered is between 82% and 88%. This corresponds to the fraction of population which was infected.
THE MODEL CHECKING PROBLEM

Given a MiTL formula $\varphi$, and a trajectory $\sigma$, we can algorithmically check if $\varphi$ holds in the world described by $\sigma$ (starting at time zero). In this case, we say that $\sigma$ is a model of $\varphi$, and we write $\sigma \models \varphi$.

A stochastic model defines a probability distribution on the space of trajectories. In this case, given a MiTL formula $\varphi$, we are interested in the probability of the subset of trajectories that satisfies $\varphi$:

$$p(\varphi) = p\{X = \sigma \mid \sigma \models \varphi\}$$

The stochastic model checking problem is thus the problem of computing $p(\varphi)$, for a given stochastic process $X(t)$. 

We start with a CTMC with fixed parameters $\theta$ and a MiTL formula $\varphi$.

We generate samples of the Bernoulli random variable $Z_{\varphi}$, equal to 1 if and only if $\varphi$ is true as follows:

- generate a sample trajectory of the CTMC, e.g. by SSA algorithm
- run a monitoring algorithm on the trajectory to establish the truth of $\varphi$.

Use statistical tools (Wald sequential testing, Bayesian approach) to establish if $p(\varphi|\theta) > q$ is true (with confidence level $\alpha$) or to estimate $p(\varphi|\theta)$. 
Outline

1. Introduction
2. Background
3. Smoothed Model Checking
4. Robust Design
5. Conclusions
Uncertainty and CTMC

Uncertain CTMC
Consider a Population CTMC model $\mathcal{M}$ depending on a set of $d$ parameters $\theta$. We only know $\theta \in \mathcal{D}$, but not their precise value. We call $\mathcal{M}$ an Uncertain CTMC (notation: $\mathcal{M}_\theta$ for fixed $\theta$).

Satisfaction function
Suppose we have a UCTMC $\mathcal{M}$, $\theta \in \mathcal{D}$, and a Metric Temporal Logic formula $\varphi$. The satisfaction probability of $\varphi$ w.r.t. $\mathcal{M}$ is a function of $\theta$:

$$p(\varphi \mid \theta) = \text{Prob}\{\mathcal{M}_\theta, x, 0 \models \varphi\}.$$

We call $p(\varphi \mid \theta)$ the satisfaction function.

Goal: can we estimate statistically $p_\varphi(\theta) = p(\varphi \mid \theta)$?
Key property of satisfaction function

Theorem (L.B. and G.S. 2014)

Consider a Population CTMC model $\mathcal{M}$ whose transition rates depend polynomially on a set of $d$ parameters $\theta \in \mathcal{D}$. Let $\varphi$ be a MITL formula. The satisfaction function $p(\varphi \mid \theta): \mathcal{D} \rightarrow [0, 1]$ is a smooth function of the parameters $\theta$.

This means that we can transfer information across neighbouring points.

Can we define a statistical model checking technique which simultaneously computes the whole satisfaction function?
SMOOTHED MODEL CHECKING: AN INTUITIVE VIEW
SMOOTHED MODEL CHECKING: AN INTUITIVE VIEW

probability
0

parameter

1

0
SMOOTHED MODEL CHECKING: AN INTUITIVE VIEW
Smootherd model checking: An intuition

Observations
We observe boolean values at different points of the parameter space.

They are generated by a Bernoulli (or Binomial) process: at each $\theta$, we sample the boolean $\text{true}$ with probability $p_\varphi(\theta)$.

Regression
We can learn a statistical model of the function $p_\varphi(\theta)$ given the observations, in a Bayesian way. For this, we need a prior distribution over $[0, 1]$-valued functions with domain $\mathcal{D}$.

But... how does a random function looks like?
**Random functions**

- The object we are interested in is a smooth function. How do we construct probability distributions over random functions?
- Simplest idea: fix a set of functions \( \xi_i(\theta) \quad i = 1, \ldots, N \) (basis functions)
- Take a linear combination \( f \) of the basis function with random coefficients \( w_i, f = \sum_i w_i \xi_i(\theta) \)
- If the coefficients are (multivariate) Gaussian distributed, the value of \( f \) at a point \( \hat{\theta} \) is Gaussian distributed
- By choosing suitable infinite sets of basis functions, we obtain Gaussian Processes.
A GP is a probability measure over the space of continuous functions (over a suitable input space) such that the random vector obtained by evaluating a sample function at a finite set of points follows a multivariate normal distribution.

A GP is uniquely defined by its mean and covariance functions, denoted by \( \mu(x) \) and \( k(x, x') \):

\[
f \sim \mathcal{GP}(\mu, k) \leftrightarrow f = (f(x_1), \ldots, f(x_N)) \sim \mathcal{N}(\mu, K),
\]

\[
\mu = (\mu(x_1), \ldots, \mu(x_N)), \quad K = (k(x_i, x_j))_{i,j}
\]

A kernel defines a space of functions, known as reproducing kernel Hilbert space.
The Radial Basis Function Kernel

The kernel function is the most important ingredient of GP (the prior mean function can be taken as the constant zero).

Radial Basis Function Kernel

\[ k(x, x') = \gamma \exp\left[-\frac{\|x - x'\|^2}{\lambda^2}\right] \]

It depends on two hyper-parameters, the amplitude \( \gamma \) and the lengthscale \( \lambda \). Sample functions from a GP with RBF covariance are with probability 1 smooth functions.

Universality Property of the RBF Kernel

Any smooth function can be approximated to arbitrary precision by a sample from a GP with RBF covariance.
Bayesian prediction with GPs

- If we have noisy observations \( y = y_1, \ldots, y_N \) of the function at inputs \( x = x_1, \ldots, x_N \), what can we say of the function value at a new point \( x^* \)?
- By Bayes’ theorem, we have

\[
p(y^*|y) \propto \int df(x)p(y^*, f(x))p(y|f(x)) \tag{1}
\]

where \( f(x) \) is the vector of function values at the input points.
- If \( p(y|f(x)) \) is Gaussian (say with zero mean and variance \( \sigma^2 \)), then we have a regression task and the integral in (1) can be computed analytically.
GP REGRESSION - EXAMPLE
**Smoothed model checking**

We call the following statistical model checking algorithm *Smoothed model checking*

1. Input a set of parameter values and a number of observations per parameter value
2. Perform (approximate) GP prediction to obtain estimates of satisfaction function and uncertainties (i.e. confidence bounds)
3. If uncertainty is too high, increase the resolution of the parameter grid/ number of observations and repeat.
4. Return estimated satisfaction function and uncertainties

We use GP regression on the inverse probit transform of the probability distribution, approximating the integral (1) by Expectation-Propagation. In fact, $p(y|f(x))$ is Binomial and the integral cannot be computed analytically.
**Network Epidemics**

SIR epidemic spreading model, with 100 nodes in one of three states: susceptible \((X_S)\), infected \((X_I)\), and recovered \((X_R)\).

**External Infection:** \(S \xrightarrow{k_e} I\), with rate function \(k_e X_S\);  
**Internal Infection:** \(S + I \xrightarrow{k_i} I + I\), with rate function \(k_i X_S X_I\);  
**Patching:** \(I \xrightarrow{k_r} R\), with rate function \(k_r X_I\);  
**Immunity Loss:** \(R \xrightarrow{k_s} S\), with rate function \(k_s X_R\).
**Example: Epidemics**

**SIR Model**

We investigate the dependence of truth probability of the property

\[ \varphi = (X_I > 0) U_{[100,120]} (X_I = 0) \]

on the infection rate: \( k_I \) and recovery rate: \( k_R \). We use 10 simulations at each of 200 parameter values = 2000 runs.

(Left: function of \( k_I \), right: function of \( k_R \). Blue dots, SMC for 5000 runs per parameter value)
**Example: Epidemics**

**SIR model**

We estimate \( p(\varphi \mid k_I, k_R) \) from **2560 traces** for the same property.

\[
\varphi = (X_I > 0) \mathbf{1}_{[100,120]}(X_I = 0)
\]

(left: smoothed MC, right SMC on 16x16 points at 5000 runs per point. Time SMC= 580× smoothed MC)
**Example: Epidemics**

<table>
<thead>
<tr>
<th>Obs. per value</th>
<th>Bayesian SMC</th>
<th>RMSE</th>
<th>Smoothed MC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64 points</td>
<td>100 points</td>
<td>256 points</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.1251 ± 0.013</td>
<td>0.0755 ± 0.036</td>
<td>0.0591 ± 0.020</td>
<td>0.0362 ± 0.014</td>
</tr>
<tr>
<td>10</td>
<td>0.0887 ± 0.010</td>
<td>0.0513 ± 0.022</td>
<td>0.0444 ± 0.023</td>
<td>0.0283 ± 0.007</td>
</tr>
<tr>
<td>20</td>
<td>0.0619 ± 0.006</td>
<td>0.0321 ± 0.016</td>
<td>0.0302 ± 0.011</td>
<td>0.0202 ± 0.005</td>
</tr>
<tr>
<td>50</td>
<td>0.0393 ± 0.006</td>
<td>0.0225 ± 0.008</td>
<td>0.0194 ± 0.004</td>
<td>0.0154 ± 0.001</td>
</tr>
<tr>
<td>100</td>
<td>0.0269 ± 0.003</td>
<td>0.0190 ± 0.004</td>
<td>0.0162 ± 0.002</td>
<td>0.0140 ± 0.002</td>
</tr>
<tr>
<td>200</td>
<td>0.0198 ± 0.002</td>
<td>0.0174 ± 0.006</td>
<td>0.0146 ± 0.003</td>
<td>0.0100 ± 0.003</td>
</tr>
</tbody>
</table>

- SmoothedMC produces consistently better estimates in the sampled grid points that Bayesian SMC, for the same number of samples.
- It also provides estimates with roughly the same standard deviation in all other points of the parameter space.
**Running times**

We report running times for another example (LacZ operon), for which simulation is more expensive, on a 2-dimensional parameter space.

<table>
<thead>
<tr>
<th>Method</th>
<th>100 points</th>
<th>256 points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smoothed MC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hyperparam. Opt.</td>
<td>8 sec</td>
<td>30 sec</td>
</tr>
<tr>
<td>GP Prediction</td>
<td>1 sec</td>
<td>2 sec</td>
</tr>
<tr>
<td>Total</td>
<td>40 sec</td>
<td>142 sec</td>
</tr>
<tr>
<td>SMC (10 runs)</td>
<td>31 sec</td>
<td>110 sec</td>
</tr>
<tr>
<td>Bayesian SMC (100 runs)</td>
<td></td>
<td>1100 sec</td>
</tr>
</tbody>
</table>
U-check: a tool for statistical model checking of uncertain CTMC

- Java-based tool
- Command line interface
- Experiment configuration file

**INPUT:** Model specified in PRISM, Bio-PEPA, ...

**INPUT:** Properties specified in MITL.

**INPUT:** Analysis mode and options

It supports smoothed Model Checking, robust parameter synthesis and parameter estimation.

https://github.com/dmilios/U-check
http://homepages.inf.ed.ac.uk/dmilios/uchek/
OUTLINE

1. Introduction
2. Background
3. Smoothed Model Checking
4. Robust Design
5. Conclusions
Robust Design with Temporal Logic

Qualitative data
We consider a set of system requirements specified as (linear) temporal logic formulae.

Problem: Robust Synthesis/Design
Find the set of model parameters that such that the model satisfies the requirements as robustly as possible.

Notion of robustness
This requires a proper notion of robustness of satisfaction for a temporal logic formula.
Signal Temporal Logic

STL is metric linear time logic for real-valued signals.

**STL Syntax**

Given a (primary) real-valued signal $x[t] = (x_1[t], ..., x_n[t])$, $t \in \mathbb{R}_{\geq 0}$, $x_i \in \mathbb{R}$, the *STL syntax* is given by

$$\varphi := \mu \mid \neg \varphi \mid \varphi_1 \land \varphi_2 \mid \varphi_1 \mathbf{U}_{[a,b]} \varphi_2$$

- $\mu : \mathbb{R}^n \to \mathbb{B}$ is an atomic predicate s.t. $\mu(x) := (y(x) \geq 0)$,
- $y : \mathbb{R}^n \to \mathbb{R}$ a real-valued function, the secondary signal.

As usual, $\mathbf{F}_{[a,b]} \varphi := \top \mathbf{U}_{[a,b]} \varphi$ and $\mathbf{G}_{[a,b]} \varphi := \neg \mathbf{F}_{[a,b]} \neg \varphi$. 
STL Quantitative Semantics

The quantitative satisfaction function $\rho$ is defined by

$$\rho(\mu, x, t) = y(x[t]) \quad \text{where} \quad \mu \equiv (y(x[t]) \geq 0)$$

$$\rho(\neg \varphi, x, t) = -\rho(\varphi, x, t)$$

$$\rho(\varphi_1 \land \varphi_2, x, t) = \min(\rho(\varphi_1, x, t), \rho(\varphi_2, x, t))$$

$$\rho(\varphi_1 \mathbf{U}_{[a,b]} \varphi_2, x, t) = \max_{t' \in t+[a,b]} \left( \min_{t'' \in [t,t']} \rho(\varphi_2, x, t') \right), \min_{t'' \in [t,t']} \rho(\varphi_1, x, t'')$$

This satisfaction score can be computed efficiently for piecewise linear signals, see e.g. the Breach Matlab Toolbox
Robustness of Stochastic Models

The quantitative satisfaction function

$$\rho(\varphi, \mathbf{x}) : \mathcal{D} \rightarrow \mathbb{R}$$

induces a real-valued random variable $$R_\varphi(\mathbf{X})$$ with probability distribution

$$\mathbb{P}(R_\varphi(\mathbf{X}) \in [a, b]) = \mathbb{P}(\mathbf{X} \in \{\mathbf{x} \in \mathcal{D} \mid \rho(\varphi, \mathbf{x}, 0) \in [a, b]\})$$

Indicators:

- $$\mathbb{E}(R_\varphi)$$ (The average robustness degree)
- $$\mathbb{E}(R_\varphi \mid R_\varphi > 0)$$ and $$\mathbb{E}(R_\varphi \mid R_\varphi < 0)$$ (The conditional average)
**Schlögl System**

CTMC model of a Schlögl system.

Biochemical reactions of the Schlögl model:

<table>
<thead>
<tr>
<th>Reaction</th>
<th>rate constant</th>
<th>init pop</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A + 2X \rightarrow 3X$</td>
<td>$k_1 = 3 \cdot 10^{-7}$</td>
<td>$X(0) = 247$</td>
</tr>
<tr>
<td>$3X \rightarrow A + 2X$</td>
<td>$k_2 = 1 \cdot 10^{-4}$</td>
<td>$A(0) = 10^5$</td>
</tr>
<tr>
<td>$B \rightarrow X$</td>
<td>$k_3 = 1 \cdot 10^{-3}$</td>
<td>$B(0) = 2 \cdot 10^5$</td>
</tr>
<tr>
<td>$X \rightarrow B$</td>
<td>$k_4 = 3.5$</td>
<td></td>
</tr>
</tbody>
</table>

Simulation of the Schlogl model (100 runs):

starting close to the boundary of the basin of attraction, the **bistable** behaviour is evident
**Schlögl System**

STL formula:

\[ \varphi : F_{[0,T_1]} G_{[0,T_2]}(X \geq k_t) \quad k_t = 300 \]

Statistical estimation of \( \varphi \): \( p = 0.4583 \) (10000 runs, error \( \pm 0.02 \) at 95\% confidence level).
SCHLÖGL SYSTEM

Satisfaction probability versus average robustness degree

Varying threshold $k_t$

Correlation around 0.8386, dependency seems to follow a sigmoid shaped curve.

Varying $k_3$

Correlation around 0.9718 with an evident linear trend.
THE SYSTEM DESIGN PROBLEM

“Given a stochastic model, depending on a set of parameters \( \theta \in K \), and a specification \( \varphi \) (STL formula), find the parameter combination \( \theta^* \) s.t. the system satisfies \( \varphi \) as robustly as possible”.

Solution strategy:

- **rephrase** it as an unconstrained optimisation problem: we maximise the expected robustness degree.
- **evaluate** the function to optimise using statistical model checking with a fixed number of runs
- **solve** the optimisation problem using the Gaussian Process-Upper Confidence Bound optimisation (GP-UCB)
**The optimisation problem**

- We need to maximise the expected robustness.
- Each evaluation of these function is **costly** (we obtain it by SMC, need to run SSA many times).
- Each evaluation of these functions is **noisy** (we estimate the value by SMC - noise is approximatively gaussian).

We need to maximise an unknown function, which we can observe with noise, with the minimal number of evaluations.

We use the GP-UCB algorithm (Srinivas et al 2012)
The GP-UCB Algorithm

**Basic Idea**

Use GP regression to emulate the unknown function, and to explore the region near the maximum of the posterior mean. Doing this naively ⇒ trapped in local optima.

**Balance Exploration and Exploitation**

Maximise an upper quantile of the distribution, obtained as mean value plus a constant times the standard deviation:

\[
x_{t+1} = \arg\max_x \left[ \mu_t(x) + \beta_t \sqrt{\text{var}_t(x)} \right]
\]

Then \(x_t\) is added to the observation points.

The algorithm has a convergence guarantee in terms of regret bounds (for slowly increasing \(\beta_t\)).
THE GP-UCB ALGORITHM - EXAMPLE
The GP-UCB Algorithm - Example
THE GP-UCB ALGORITHM - EXAMPLE
The GP-UCB Algorithm - Example
Experimental Results (Schlögl System)

Statistics of the results of ten experiments to optimize the parameter $k_3$, for $\varphi : F^{[0,T_1]} G^{[0,T_2]} (X \geq k_t)$, in the range $[50, 1000]$:

<table>
<thead>
<tr>
<th>Parameter mean</th>
<th>Parameter range</th>
<th>Mean probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_3 = 997.78$</td>
<td>$[979.31 \ 999.99]$</td>
<td>1</td>
</tr>
<tr>
<td>Average Robustness</td>
<td>Number of function evaluations</td>
<td>Number of simulation runs</td>
</tr>
<tr>
<td>348.97</td>
<td>34.4</td>
<td>3440</td>
</tr>
</tbody>
</table>

The emulated robustness function in the optimisation of $k_3$
**Experimental Results (Schlögl System)**

The distribution of the robustness score for $\varphi : F^{[0,T_1]} G^{[0,T_2]} (X \geq k_t)$ with $k_3 = 999.99$, $T_1 = 10$, $T_2 = 15$ and $k_t = 300$
OUTLINE

1. INTRODUCTION
2. BACKGROUND
3. SMOOTHED MODEL CHECKING
4. ROBUST DESIGN
5. CONCLUSIONS
Conclusions

- Uncertainty has implications also for formal analysis techniques.
- When there is uncertainty, machine learning is likely to be of use.
- We proposed efficiently methods relying on Gaussian Processes for the satisfaction under uncertainty, system design, parameter estimation from qualitative observations, ... 
- We have a working tool: U-check.
- Challenges: uncertain model structure, model abstraction, modularisation, spatio-temporal models.
ACKNOWLEDGEMENTS

SPONSORS
FP7 QUANTICOL EU project and UniTS local funding.

COLLABORATORS
- Guido Sanguinetti
- Dimitios Milios
- Laura Nenzi
- Ezio Bartocci
- Tomas Bradzil
- Giulio Caravagna
QEST 2017 (ADV)

14th Int. Conference on Quantitative Evaluation of SysTems
Berlin, 5-7 September 2017

Deadline: 31 March 2017
The End
Smoothed Model Checking, More Formally

- At each $\theta$, the distribution of observations is $\text{Binomial}(m, p_\phi(\theta))$.

- We collect observations at input points $\theta_i$ in a matrix $\mathcal{D}$.

- The inverse probit transform of the satisfaction function $g_\phi(\theta) = \psi(p_\phi(\theta))$ (a smooth function of the parameters) is assigned a GP prior. Here

$$
\psi(p) = g \iff p = \int^g \mathcal{N}(0, 1) \quad \forall p \in [0, 1], g \in \mathbb{R}
$$

- $g^* = g_\phi(\theta^*)$ at a new point $\theta^*$, given observations $\mathcal{D}$, is then distributed as

$$
p(g_\phi(\theta^*)|\mathcal{D}) \propto \int dg \mathcal{N}((g^*, g)|0, \Sigma) \prod_{i=1}^{N} (p_\phi(\theta_i))^{\sum d_i} (1 - p_\phi(\theta_i))^{m - \sum d_i}
$$

- An analytical approximation of this integral can be computed by a variant of the Expectation-Propagation algorithm.