Model Checking Adversarial Robustness in Stochastic Systems

Authors
Lisa Oakley, Alina Oprea, Stavros Tripakis, Northeastern University, Boston, MA

Contact
oakley.l@northeastern.edu
https://lisaoakley.github.io/model-checking-robustness

Abstract

Probabilistic model checking is a useful technique for specifying and verifying stochastic systems. These methods typically rely on the assumed structure and probabilities of certain system transitions which may be violated by adversarial manipulation. We develop a formal definition of adversarial robustness and a flexible framework for modeling adversaries in discrete time Markov chains (DTMCs). We develop attack synthesis algorithms and evaluate our methods on a set of DTMC case studies.

DTMC Adversarial Robustness

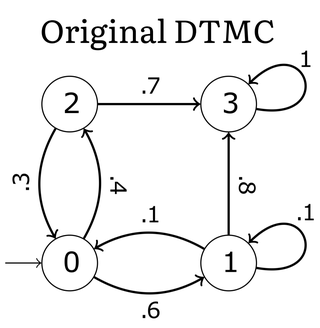
Given $0 \leq \delta \leq 1$, DTMC $\mathcal{C} = (\mathcal{S}, s_0, \mathbf{P})$, perturbation set PS, and some $PCTL^*$ path formula φ, $\mathcal{C}$ is *adversarially robust* with respect to PS, φ, δ if

$$Pr(s_0 \models_{\mathcal{C}'} \varphi) \geq Pr(s_0 \models_{\mathcal{C}} \varphi) - \delta$$

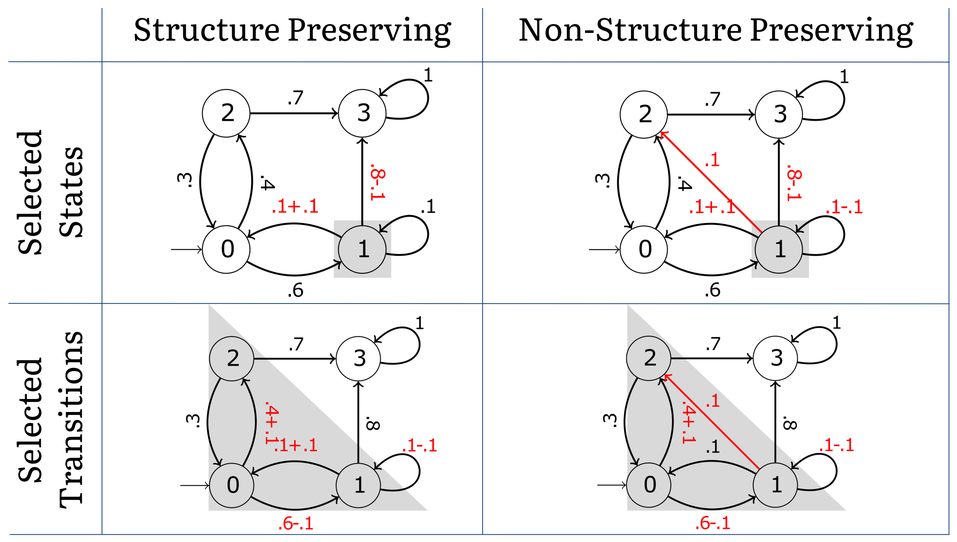
for all $\mathcal{C}' \in PS$ where PS is a set of DTMCs of the form $(\mathcal{S}, s_0, \mathbf{P}')$.

Modeling the Adversary

Our adversarial robustness framework models an adversary as a set (PS) of possible perturbations to the DTMC transition probabilities. In this work, we consider a class of adversarial models for which PS is a set of perturbations within an epsilon ball around the original transition probabilities with respect to the max distance function between the perturbation matrices: $\|\mathbf{P} - \mathbf{P}'\|_{max} \leq \varepsilon$. We show four specific instances of these threat models.



$$\mathbf{P} = \begin{bmatrix} 0 & .6 & .4 & 0 \\ .2 & 0 & 0 & .8 \\ .3 & 0 & 0 & .7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Optimization Solutions

We model the specific attacker as a matrix $\mathbf{X}$ of transition probability perturbations. We set up an optimization problem to find the $\mathbf{X}$ which minimizes the probability that the perturbed DTMC (with transition probability matrix $\mathbf{P}$+$\mathbf{X}$) satisfies a temporal logic property with respect to the given attacker bounds. Formally,

Given DTMC $\mathcal{C} = (\mathcal{S}, s_0, \mathbf{P})$, perturbation set PS, and some PCTL* path formula φ, find a perturbation matrix $\mathbf{X}^*$ which solves

$$\underset{\mathbf{X}}{argmin}\ Pr(s_0 \models_{\mathcal{C}'} \varphi)$$
$$subject\ to\ \mathcal{C}' = (\mathcal{S}, s_0, \mathbf{P} + \mathbf{X}) \in PS$$

DTMC $\mathcal{C}$ is adversarially robust if the probability of satisfying the property under worst-case attack is within the δ threshold.

We propose and implement two methods to solve this optimization problem:

1) Direct objective computation using the PRISM[1] probabilistic model checker.
2) Pre-computing a symbolic objective function using the PARAM [2] parametric modeling tool.

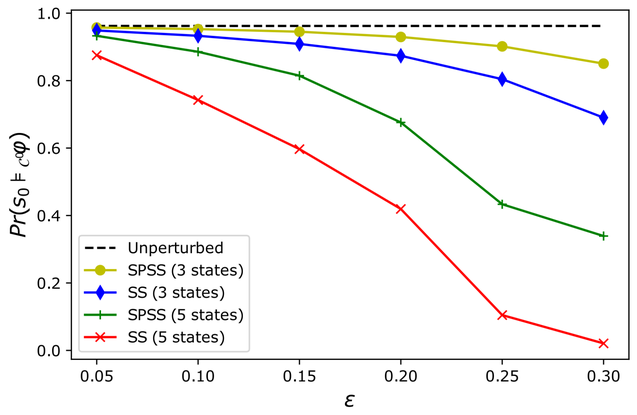
We implement the experiments in Python and use the SciPy Sequential Least Squares Programming (SLSQP) minimizer.

Conclusion

Prior work on perturbation analysis in DTMCs focuses primarily on perturbations which maintain the structure of the DTMC, and does not allow for analysis of specific attackers. In our work, we develop a formal model for reasoning about large classes of adversaries. We develop two solutions which finds the worst-case attack on a system with respect to a perturbation set and temporal logic property. We evaluate our solution and provide examples of potential uses for our framework in system analysis.

Results

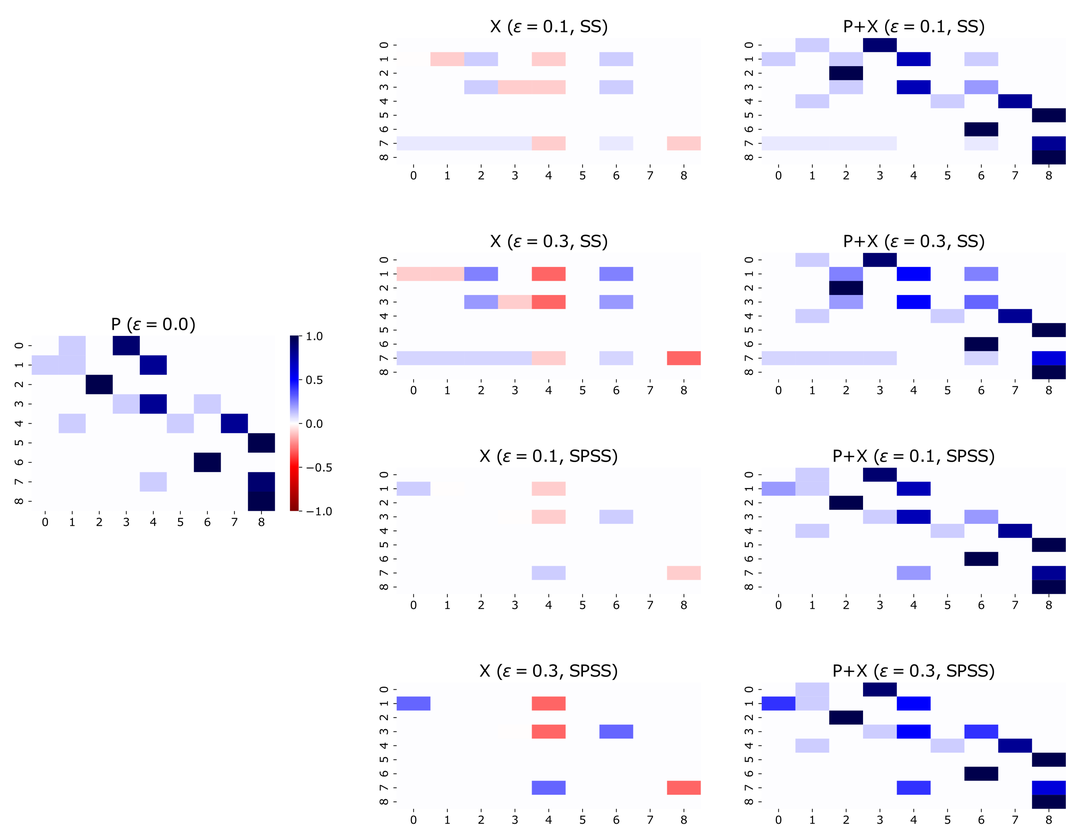
We measure the effect that the amount of perturbation has on the probability that the DTMC reaches a success state within 200 time steps for a 25 state Gridworld DTMC. We compare structure preserving and non-structure preserving selected states threat models with 3 and 5 vulnerable states, with perturbation budget ε ranging from 0.05 to 0.3.We see that the attacks which can modify the structure of the DTMC have a greater impact on robustness.



We compare the two solution methods over randomized 5×5, 10×10, and 15×15 DTMCs using a selected transitions threat model with 5, 10, and 20 vulnerable transitions (parameters) to illustrate the relative growth in computation time between the two solutions.

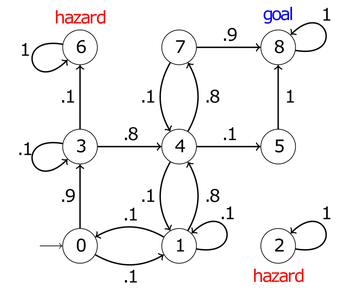
We see that precomputing the symbolic solution leads to significantly faster optimization times (right hand side of the addition) compared to using PRISM [1] to directly compute the objective at every step of the optimization. However, precomputing the solution function quickly becomes expensive especially with larger state spaces, and leads to a timeout when the state size grows beyond 100.

Property	# States	# Params	Method	Total Duration (in seconds)
P=? [s!=5 U s=24]	25	5	Direct Computation	**0.354**
P=? [s!=5 U s=24]	25	5	Symbolic Soln. Func.	0.052 + 0.076 = **0.128**
P=? [s!=5 U s=24]	25	10	Direct Computation	**2.445**
P=? [s!=5 U s=24]	25	10	Symbolic Soln. Func.	124.316 + 0.98 = **125.296**
P=? [s!=5 U s=24]	25	20	Direct Computation	**4.884**
P=? [s!=5 U s=24]	25	20	Symbolic Soln. Func.	TO
P=? [s!=10 U s=99]	100	5	Direct Computation	**12.19**
P=? [s!=10 U s=99]	100	5	Symbolic Soln. Func.	2.006 + 1.089 = **3.095**
P=? [s!=10 U s=99]	100	10	Direct Computation	**23.69**
P=? [s!=10 U s=99]	100	10	Symbolic Soln. Func.	717.126 + 2.792 = **719.918**
P=? [s!=10 U s=99]	100	20	Direct Computation	**67.616**
P=? [s!=10 U s=99]	100	20	Symbolic Soln. Func.	TO
P=? [s!=15 U s=224]	225	5	Direct Computation	**59.32**
P=? [s!=15 U s=224]	225	5	Symbolic Soln. Func.	TO
P=? [s!=15 U s=224]	225	10	Direct Computation	**134.714**
P=? [s!=15 U s=224]	225	10	Symbolic Soln. Func.	TO
P=? [s!=15 U s=224]	225	20	Direct Computation	**290.201**
P=? [s!=15 U s=224]	225	20	Symbolic Soln. Func.	TO



We visualize the DTMC transition probability matrices and worst-case attacker perturbation matrices for a 3×3 Gridworld DTMC (pictured below). We consider structure preserving and non-structure preserving selected states threat models with vulnerable states 1, 3, and 7, with respect to the probability that the system reaches the goal, while avoiding the hazards, in 6 steps.

Attacker perturbations increase probability of transitioning to hazard states and decrease probability of transitioning to the goal state. In the non-structure preserving case, the previously unreachable hazard (state 2) becomes reachable after attack. This sort of analysis can be useful to a system designer, as it illuminates not only the effects of attacks on the system, but the specific components which are most vulnerable.



3×3 Gridworld

[1] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in Proc. 23rd International Conference on Computer Aided Verification (CAV'11), ser. LNCS, Springer, 2011
[2] E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang, "PARAM: A Model Checker for Parametric Markov Models," in Computer Aided Verification. ser. LNCS, Springer, 2010

This work has been supported by the National Science Foundation under NSF SaTC awards CNS-1717634 and CNS-1801546.