Input Attribution for Statistical Model Checking using Logistic Regression

Jeff Hansen

September 23, 2016

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM-0004023

Motivating Example

Pursuer/Evader Example

- Pursuer and Evader given random initial positions (x_p, y_p) and (x_e, y_e) near center of region.
- Evader attempts to reach safe zone in corner.
- Faster moving pursuer attempts to catch evader before it reaches safe zone.

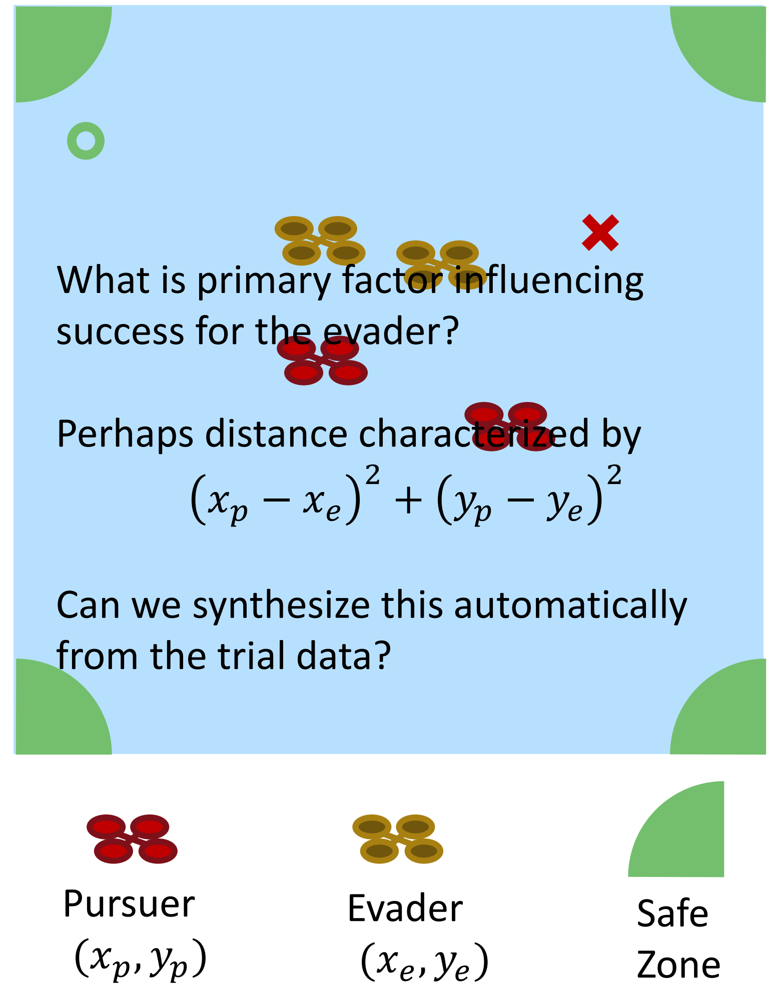
Statistical Model Checking (SMC)

- Let $\mathcal{M}$ be the model for the pursuer/evader scenario and Φ be the property "the evader reaches safe zone".
- SMC attempts to answer the question, "What is the probability that $\mathcal{M} \vDash \Phi$?"

Input Attribution (IA)

- Asks the question "Why do I get a particular probability estimate?"
- Analog to counter-example in model checking.
- Expressed in terms of the inputs as approximation for the model?

Pursuer/Evader Example



Statistical Model Checking (SMC)

$$\textbf{Relative Error} = \frac{\textbf{Std. Dev.}}{\textbf{Mean}}$$

Any system $\mathcal{M}$ that takes random inputs

Probabilistic Temporal Logic Formula $\boldsymbol{\phi}$

Statistical Model Checker

Estimated Probability that $\mathcal{M} \vDash \boldsymbol{\phi}$ with relative error $\boldsymbol{RE}$

Based on Monte-Carlo Simulation

- System properties described in formal language (UTSL, BLTL, etc.)
- Property is tested on “sample trajectories” (sequence of states)
- Each outcome can be treated as a Bernoulli random variable (i.e., coin flip)

Statistical Model Checking with Crude Monte-Carlo

The probability that condition Φ holds in model $\mathcal{M}$ when the input $\vec{x}$ is distributed according to joint pdf $f(\vec{x})$ is the expected value of that indictor function and can be calculated as:

$$p = E[I_{\mathcal{M}\vDash\Phi}(\vec{x})] = \int I_{\mathcal{M}\vDash\Phi}(\vec{x}) f(\vec{x}) d\vec{x}$$

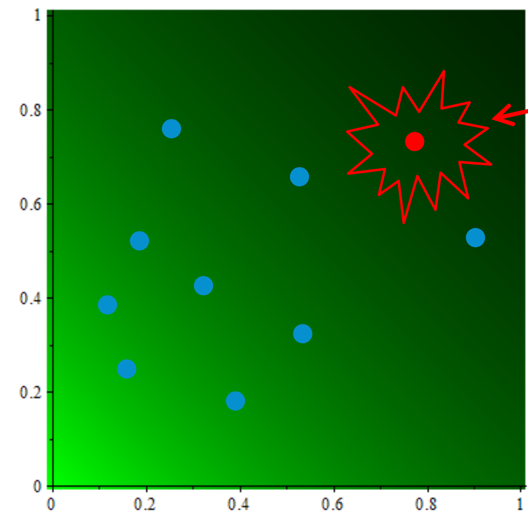
where $I_{\mathcal{M}\vDash\Phi}(\vec{x})$ is an indicator function for the model. This can be estimated with Crude Monte-Carlo simulation as:

$$\hat{p} = \frac{1}{N}\sum_{i=1}^{N} I_{\mathcal{M}\vDash\Phi}(\vec{x}_i)$$

where each $\vec{x}_i$ is a sample vector drawn from $f(\vec{x})$. As N gets large, $\hat{p}$ will converge to p.

For low probabilities, approximate number of samples required to evaluate to a relative error of RE is: $\boldsymbol{N \approx \frac{1}{p(RE)^2}}$

Estimated Failure Probability

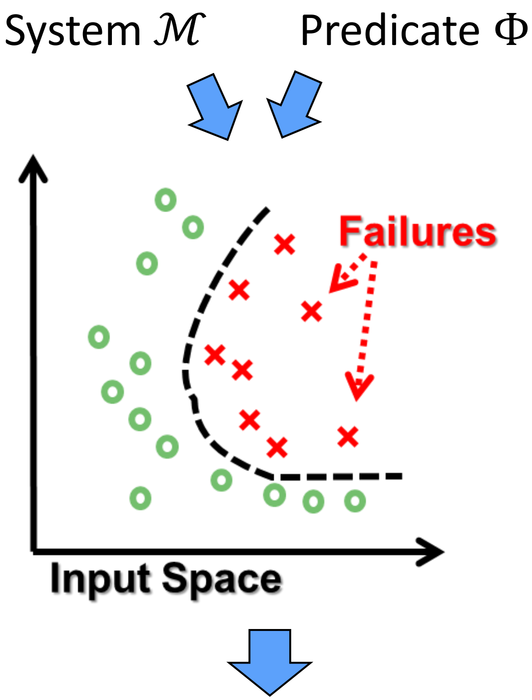


# of samples in fault region

$$\hat{p} = \frac{1}{10} = 0.1$$

total # of samples

Input Attribution – The "Why" of SMC



Input Attribution

Expression	p-Value
$0.62(a - 1.01d)^2$	0.0013
$4.3b$	0.0042
$1.3(2.3 - c)$^2	0.0172

Problem – Standard SMC provides an estimate on probability that a predicate is satisfied, but does not address why a particular result was obtained.

Goal – Provide investigator with informative non-redundant representation of how system inputs relate to the property being tested:

1. Describes relationship that actually exists in data
2. Is presented in a way that is quantitative and understandable
3. Gives investigator new insights
4. Is resilient to randomness in the system

Approach – Apply machine learning and feature extraction techniques.

- Use *Logistic Regression* to identify "predictors" that affect the probability that a predicate is satisfied.
- Calculate p-values for predictors to indicate significance.
- Look for sets of predictors that can be factored into larger expressions.

Odds vs Probability

Logistic Regression reasons about "odds"

- Alternate representation of probability
- Think of horse racing odds like "7:1".
- The odds γ of event is related to the probability p as:

$$p = \frac{\gamma}{\gamma + 1} = \frac{1}{1 + 1/\gamma}$$

- Odds fall in interval:

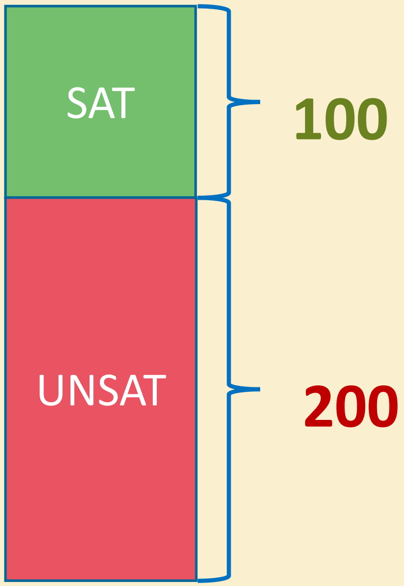
$$0 < \gamma < \infty$$

- Log of odds fall in interval:

$$-\infty < \log(\gamma) < \infty$$

- Unbounded nature of "log odds" makes it suitable for linear regression analysis.

Example



Odds

$$\gamma_{SAT} = \frac{100}{200} = 0.5$$

Probability:

$$p_{SAT} = \frac{100}{100 + 200} = 0.333$$

Logistic Regression (LR)

Logistic Regression

- Regression model useful when dependent variable is Boolean.
- Based on the logistic function.
- Linear fit of the log of the “odds”.
- Estimates probability that for a particular input the result variable will be true.

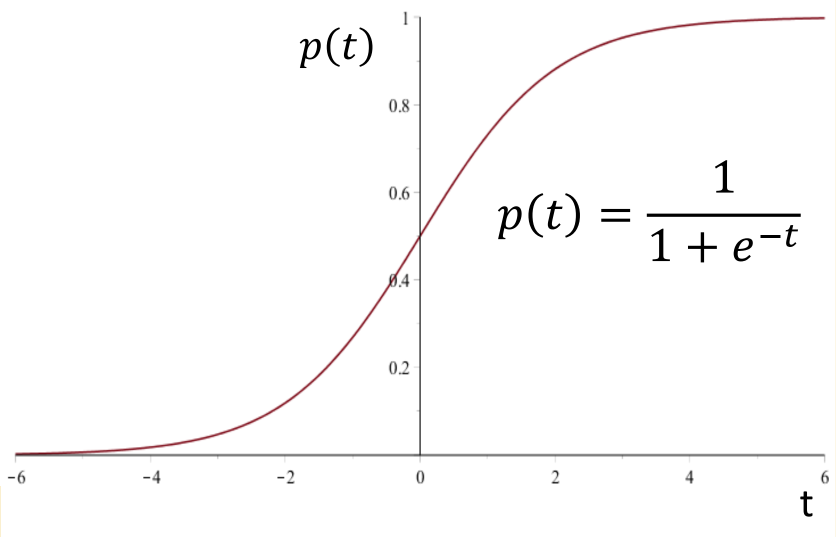
Input

- Set of “trials” consisting of vectors of “predictors” (e.g., input variables) and a dependent Boolean random variable.

Output

- Set of coefficients for each predictor that fit a linear expression.
- Standard error for each predictor from which a p-Value can be computed.

Logistic Function



$$p(t) = \frac{1}{1 + e^{-t}}$$

Log Odds or “Logit”

$$t = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_N x_N$$

LR Model

$$L(x) = \frac{1}{1 + e^{-\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_N x_N}}$$

Evaluating LR Results (Linear Case)

The factor by which the log odds of a predicate will increase per unit increase of input. Positive values represent increase and negative values represent decrease.

- Calculated by applying inverse normal distribution to ratio of standard error and β.
- Represents probability that $\beta = 0$ can explain the data.
- Values above a threshold (e.g., 0.05) indicate relation between input and predicate is not statistically significant.

Constant Term

Predictors

Name	β	Std.Err.	p-Value
–	-4.28	0.874	0.0000
a	0.154	0.0138	0.0000
b	-1.91	0.3551	0.0000
c	0.0635	0.0277	0.0219
d	5.05	2.77	0.0685

Error in estimation of β.

This predictor is not statistically significant since its p-value is greater than 0.05.

Polynomial Input Attribution

Non-Linear Predictors

- By including non-linear predictors, it may be possible to find a statistically significant solution when linear only terms fail.
- In our work to date, we have focused on quadratic terms (e.g., x^2, y^2, xy)
- Higher order or non-polynomial terms could be useful for some systems.

Factoring

- Factored polynomials are easier for humans to understand.
- Since coefficients are approximated, perfect factorings may not be possible.
- Look for approximate factorings which do not adversely affect original coefficients.

Find variable pairs with squares and cross terms

Name	β	Std.Err.	p-Value
⋮	⋮	⋮	⋮
a^2	1.01	0.0148	0.0000
ab	-2.04	0.0362	0.0000
b^2	1.02	0.0193	0.0219
⋮	⋮	⋮	⋮

Complete square to create candidate factoring

$$1.01(a - 1.01b)^2$$

$$1.01a^2 - 2.04ab + 1.03b^2$$

Re-expand and accept approximation if error is within set factor of std. error.

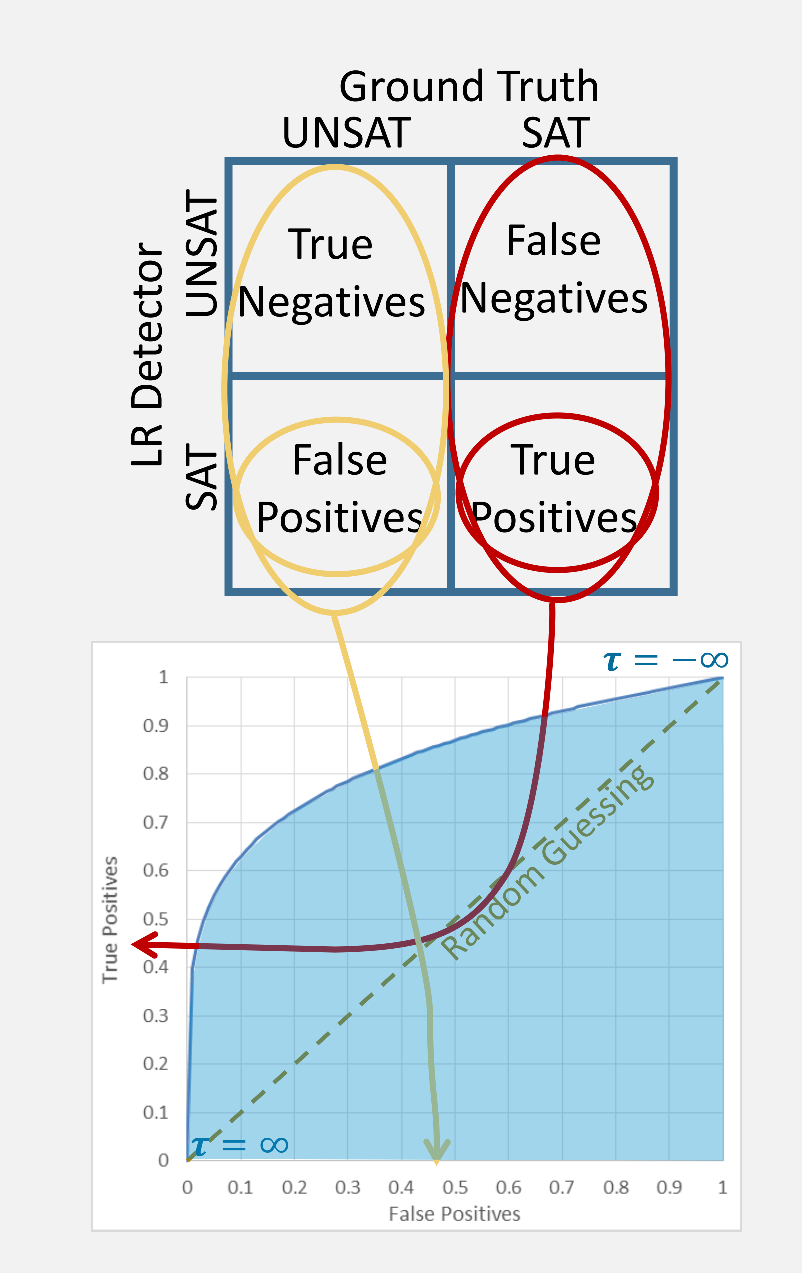
Evaluating LR Fit

Model Verification

- Results of LR only meaningful if overall model fits data.
- LR model $L(x)$ predicts probability that input x will satisfy predicate.

ROC Analysis

- ROC curve is plot of
  - true positives $\mathrm{P}[L(x_i) > \tau | \phi_i]$ vs
  - false positives $\mathrm{P}[L(x_i) > \tau | \overline{\phi_i}]$
  - for $-\infty < \tau < \infty$
- Area Under Curve (AUC)
  - Represents $\mathrm{P}[L(x_{SAT}) > L(x_{UNSAT})]$ where x_{SAT} and x_{UNSAT} are arbitrary inputs resulting in SAT and UNSAT of Φ.
  - Values range between 0.5 (model is no better than chance) to 1.0 (perfect fit).



Evaluating LR Fit

Model Verification

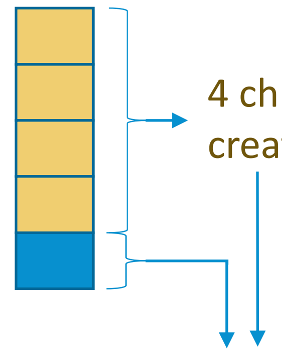
- Results of LR only meaningful if overall model fits data.
- LR model $L(x)$ predicts probability that input x will satisfy predicate.

ROC Analysis

- ROC curve is plot of
  - true positives $\mathrm{P}[L(x_i) > \tau | \phi_i]$ vs
  - false positives $\mathrm{P}[L(x_i) > \tau | \overline{\phi_i}]$
  - for $-\infty < \tau < \infty$
- Area Under Curve (AUC)
  - Represents $\mathrm{P}[L(x_{SAT}) > L(x_{UNSAT})]$ where x_{SAT} and x_{UNSAT} are arbitrary inputs resulting in SAT and UNSAT of Φ.
  - Values range between 0.5 (model is no better than chance) to 1.0 (perfect fit).
  - Use average of 5-fold cross validation to avoid bias.

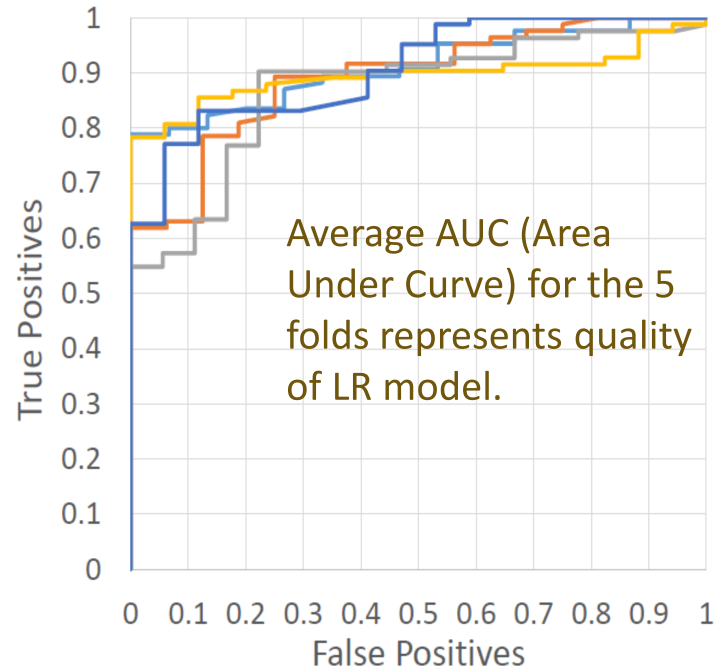
5-Fold Cross Validation

Simulation data with input and predicate results x_i, ϕ_i partitioned into 5 chunks

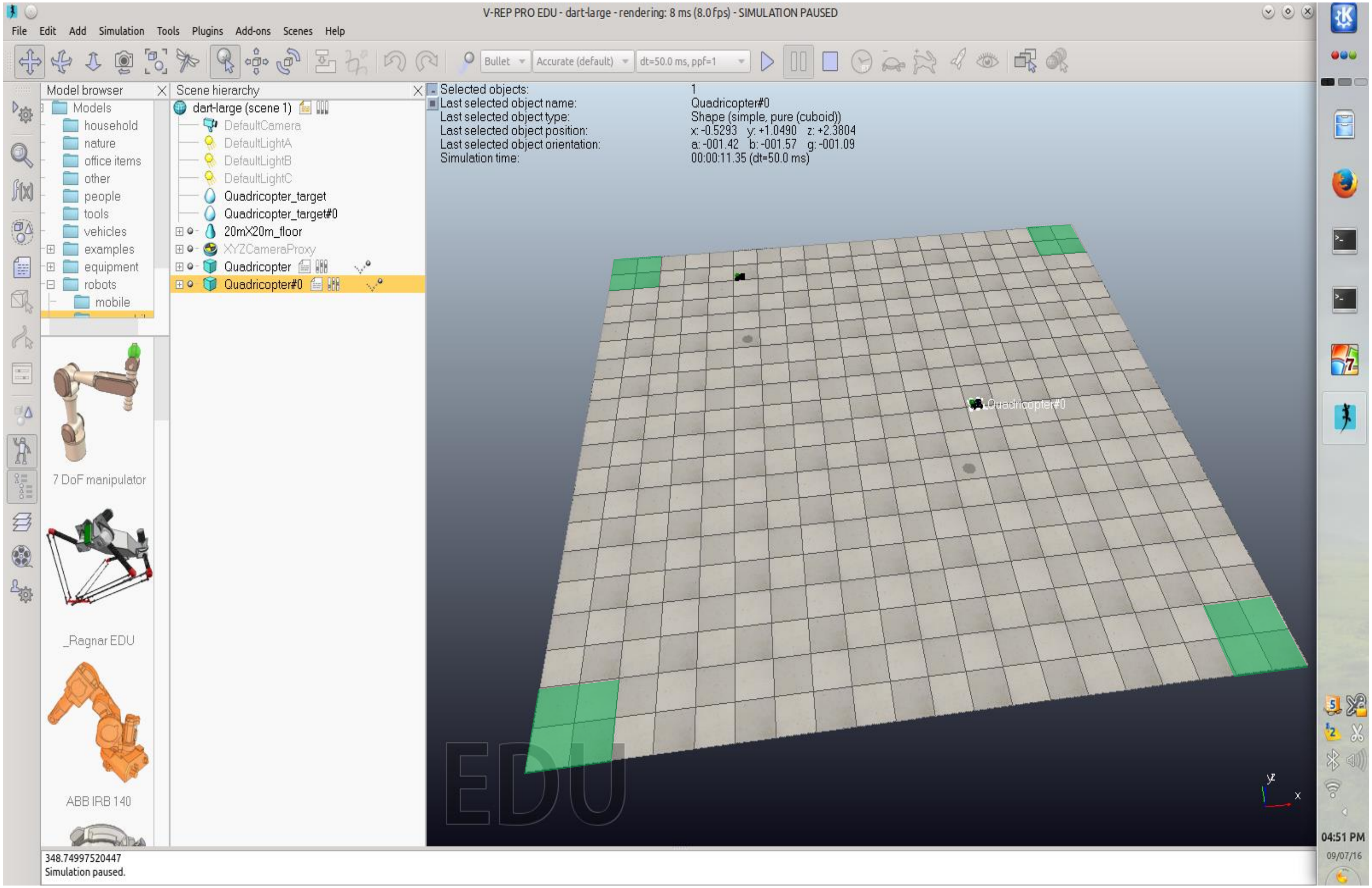


4 chunks used to create model $L(x)$

5[th] chunk compared with model to create ROC curve. (repeated for each chunk)



V-REP Simulator



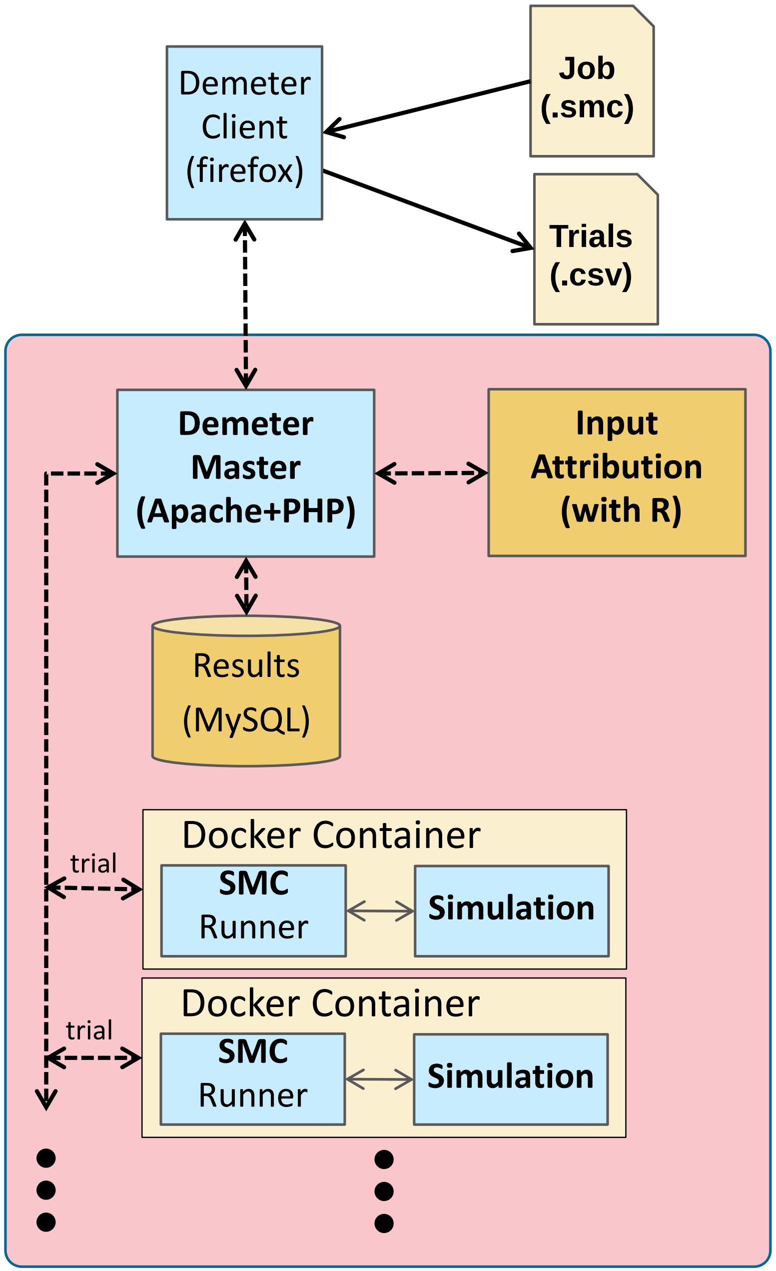
Demeter

Goal: Develop parallel infrastructure for SMC of systems with probabilistic behaviors.

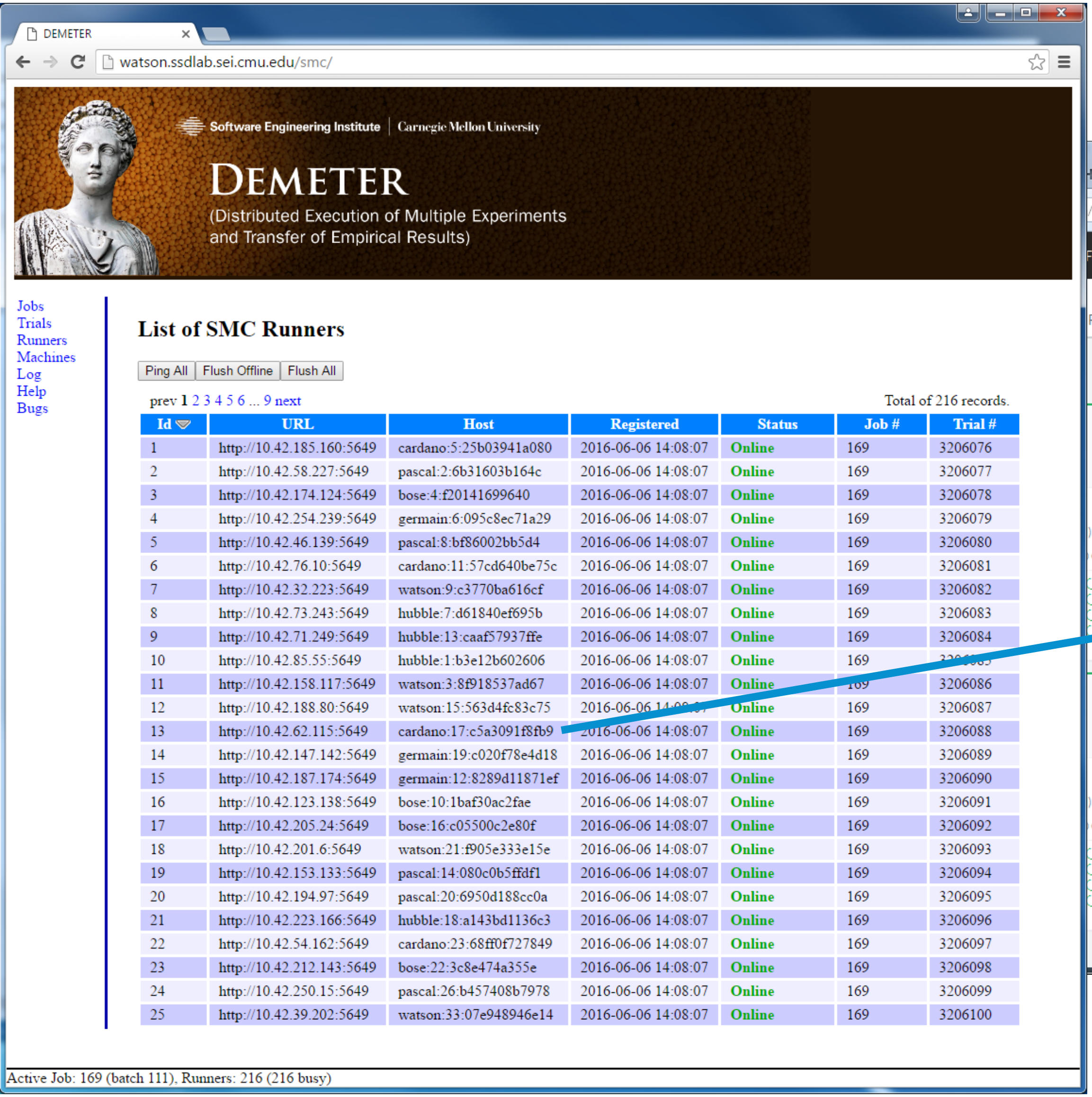
Primary target is autonomous systems.

Demeter components

- Client runs in web browser (e.g. firefox)
- Master runs in Apache server with PHP
- Data stored in MySQL database
- Input Attribution uses R statistical system
- Individual simulations conducted in Docker containers. Managed by “Runner”.



SMC-Runner operate as a Docker container. Each Docker container is managed by Rancher



DEMETER

watson.ssdlab.sei.cmu.edu/smc/

Software Engineering Institute | Carnegie Mellon University

DEMETER

(Distributed Execution of Multiple Experiments and Transfer of Empirical Results)

Jobs
Trials
Runners
Machines
Log
Help
Bugs

List of SMC Runners

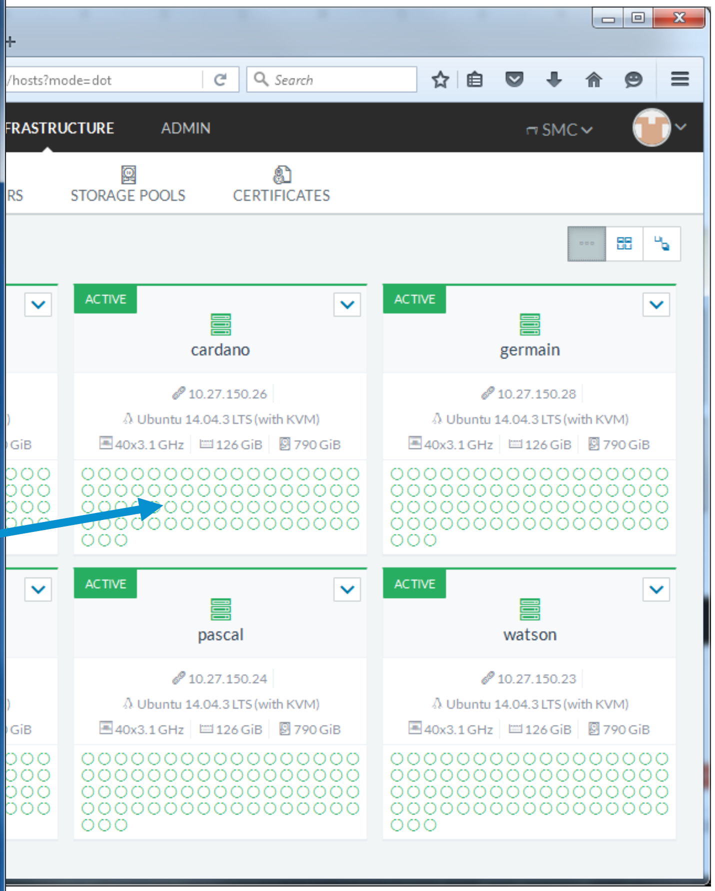
Ping All | Flush Offline | Flush All

prev 1 2 3 4 5 6 ... 9 next

Total of 216 records.

Id	URL	Host	Registered	Status	Job #	Trial #
1	http://10.42.185.160:5649	cardano:5:25b03941a080	2016-06-06 14:08:07	Online	169	3206076
2	http://10.42.58.227:5649	pascal:2:6b31603b164c	2016-06-06 14:08:07	Online	169	3206077
3	http://10.42.174.124:5649	bose:4:f20141699640	2016-06-06 14:08:07	Online	169	3206078
4	http://10.42.254.239:5649	germain:6:095c8ec71a29	2016-06-06 14:08:07	Online	169	3206079
5	http://10.42.46.139:5649	pascal:8:bf86002bb5d4	2016-06-06 14:08:07	Online	169	3206080
6	http://10.42.76.10:5649	cardano:11:57cd640be75c	2016-06-06 14:08:07	Online	169	3206081
7	http://10.42.32.223:5649	watson:9:c3770ba616cf	2016-06-06 14:08:07	Online	169	3206082
8	http://10.42.73.243:5649	hubble:7:d61840ef695b	2016-06-06 14:08:07	Online	169	3206083
9	http://10.42.71.249:5649	hubble:13:caaf57937ffe	2016-06-06 14:08:07	Online	169	3206084
10	http://10.42.85.55:5649	hubble:1:b3e12b602606	2016-06-06 14:08:07	Online	169	[illegible]
11	http://10.42.158.117:5649	watson:3:8f918537ad67	2016-06-06 14:08:07	Online	169	3206086
12	http://10.42.188.80:5649	watson:15:563d4fc83c75	[illegible]	Online	169	3206087
13	http://10.42.62.115:5649	cardano:17:c5a3091f8fb9	2016-06-06 14:08:07	Online	169	3206088
14	http://10.42.147.142:5649	germain:19:c020f78e4d18	2016-06-06 14:08:07	Online	169	3206089
15	http://10.42.187.174:5649	germain:12:8289d11871ef	2016-06-06 14:08:07	Online	169	3206090
16	http://10.42.123.138:5649	bose:10:1baf30ac2fae	2016-06-06 14:08:07	Online	169	3206091
17	http://10.42.205.24:5649	bose:16:c05500c2e80f	2016-06-06 14:08:07	Online	169	3206092
18	http://10.42.201.6:5649	watson:21:f905e333e15e	2016-06-06 14:08:07	Online	169	3206093
19	http://10.42.153.133:5649	pascal:14:080c0b5ffdf1	2016-06-06 14:08:07	Online	169	3206094
20	http://10.42.194.97:5649	pascal:20:6950d188cc0a	2016-06-06 14:08:07	Online	169	3206095
21	http://10.42.223.166:5649	hubble:18:a143bd1136c3	2016-06-06 14:08:07	Online	169	3206096
22	http://10.42.54.162:5649	cardano:23:68ff0f727849	2016-06-06 14:08:07	Online	169	3206097
23	http://10.42.212.143:5649	bose:22:3c8e474a355e	2016-06-06 14:08:07	Online	169	3206098
24	http://10.42.250.15:5649	pascal:26:b457408b7978	2016-06-06 14:08:07	Online	169	3206099
25	http://10.42.39.202:5649	watson:33:07e948946e14	2016-06-06 14:08:07	Online	169	3206100

Active Job: 169 (batch 111), Runners: 216 (216 busy)



Target/Threat Experiment

Scenario

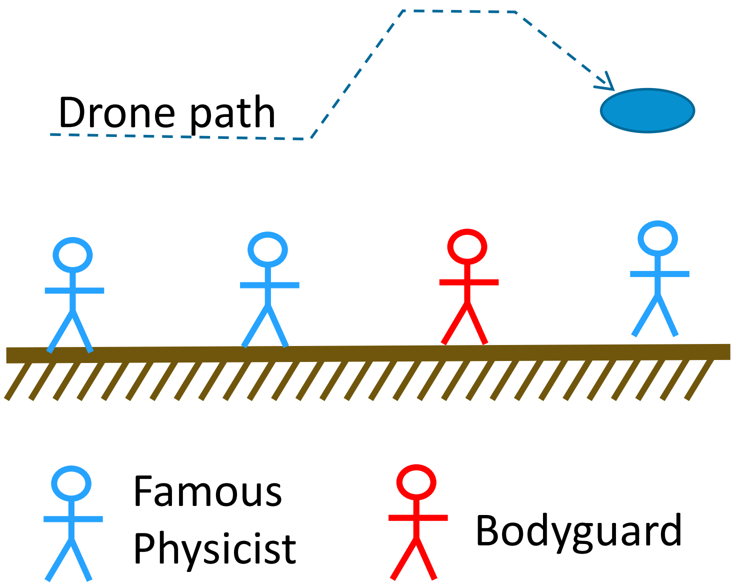
- Drone flies pre-programmed path over area.
- Along path are “targets” to be photographed.
  - Close to ground → Better chance of good photo.
- Path also includes “threats” to be avoided.
  - Close to ground → More likely to be destroyed.
- Adaptive algorithm with imperfect sensors, sense threats ahead and controls altitude.

Inputs

- Number of targets/threats
- Target detector range (down)
- Target/Threat detector range/accuracy (forward)
- Threat range

Predicate

- Drone photographs at least 50% of targets while avoiding being destroyed by threats.



Target/Threat Experiment

Key Observations

- False positives on threats reduce the probability of detecting targets.
  - Reacting to threats that are not there results in drone flying at higher altitude when not necessary and missing some targets.
- Increasing number of targets reduces probability of survival.
  - Drone takes more risks by flying lower to photograph targets.
- False negatives on threat and target detection do not have statistically significant effect on mission, detection or survival probabilities.
  - Verified with additional simulations varying false negative rate. Could indicate problem with adaptation algorithm controlling drone.

Simulation Results

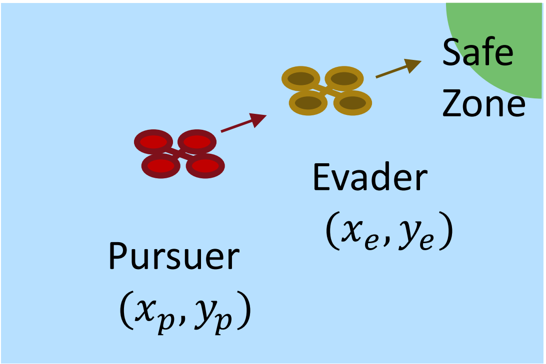
#Trials:	22,560
P[SAT-mission]:	0.308
P[SAT-survive]:	0.618
P[SAT-detect]:	0.361
Relative Error:	0.05
Batch Size:	120
Run Time:	10 hours, 6 min

Input Attribution (AUC=0.926)

Name	β mission	β detect	β survive
Target Detector Range	1.33	1.46	
Threat Range	-1.57	-1.189	-2.37
Threat Lookahead	0.233	0.194	0.377
Number of Threats	-0.0892	-0.0943	-0.0792
Number of Targets			-0.0296
Target False Positives			-17.81
Threat False Positives	-3.26	-10.04	32.74

Motivating Example - Revisited

Pursuer/Evader Example



Initial hypothesis was that initial distance between pursuer and evader was deciding factor for survival of evader.

Simulation Results

#Trials:	36,960
# SAT:	7,900
P[SAT]:	0.214
Relative Error:	0.01
Batch Size:	120
Run Time:	5 hours, 20 min

Input Attribution (AUC=0.77)

Name	β	$se(\beta)$	p-value
$x_e x_p$	-0.124	0.0027	$< 10^{-4}$
$y_e y_p$	-0.122	0.0027	$< 10^{-4}$
x_e^2	0.06	0.0031	$< 10^{-4}$
y_e^2	0.056	0.0031	$< 10^{-4}$
x_p^2	0.056	0.0031	$< 10^{-4}$
y_p^2	0.056	0.0031	$< 10^{-4}$

Factoring the IA predictors gives us:

$$0.0602(x_e - 1.03x_p)^2 + 0.0561(y_e - 1.09y_p)^2$$

With error less than $4se(\beta)$ on each coefficient.

Resulting IA expression very close to square of Euclidean distance. Constant factor represents relation between distance and log odds of survival.

Summary

Addresses the “Why” of SMC

- Concise human understandable justification for SMC probability estimate
- Modeled as expression correlated with the predicate
- Shows which variables are most important

Uses Logistic Regression

- Regression model useful when dependent variable is Boolean.
- Not dependent on input distribution.
- Commonly used in machine learning applications.

Extended to Non-Linear Attribution

- Adds non-linear functions of inputs as predictors to model.
- Can find attributions where linear attribution fails.