Towards Verifying AI Systems: Testing of Samplers

Kuldeep S. Meel

School of Computing, National University of Singapore

Joint work with Sourav Chakraborty, Indian Statistical Institute
(Relevant publication: *On Testing of Uniform Samplers*, In Proc of AAAI-19)

@FMAI 2019

The Fourth Revolution

- Andrew Ng *Artificial intelligence is the new electricity*
- Gray Scott *There is no reason and no way that a human mind can keep up with an artificial intelligence machine by 2035*
- Ray Kurzweil *Artificial intelligence will reach human levels by around 2029. Follow that out further to, say, 2045, we will have multiplied the intelligence, the human biological machine intelligence of our civilization a billion-fold.*

And yet it fails at basic tasks

- English: *Of course, I do love you. Let's have dinner this Friday? See you!*

And yet it fails at basic tasks

- English: *Of course, I do love you. Let's have dinner this Friday? See you!*
- Google translate in french: (which losely reads as follows in English) : *Of course, I do not love you. See you!*

And yet it fails at basic tasks

- English: *Of course, I do love you. Let's have dinner this Friday? See you!*
- Google translate in french: (which losely reads as follows in English) : *Of course, I do not love you. See you!*

So where are we?

- There has been a significant progress for tasks that were thought to be hard
  - Computer vision
  - Game playing
  - Machine translation

And yet it fails at basic tasks

- English: *Of course, I do love you. Let's have dinner this Friday? See you!*
- Google translate in french: (which losely reads as follows in English) : *Of course, I do not love you. See you!*

So where are we?

- There has been a significant progress for tasks that were thought to be hard
  - Computer vision
  - Game playing
  - Machine translation
- But this progress has come at the cost of understanding of how these systems actually work
- Eric Schmidt, 2015: There should be verification systems that evaluate whether an AI system is doing what it was built to do.

Imprecise systems: Adversarial Examples

(a) (b) (c) (d) (e)

(a) (b) (c) (d) (e)

- Given a model M
  - M: A neural network to label images
- Specification φ
  - φ: Label stop sign as **STOP**

The Classical Approach

- Given a model M
  - M: A neural network to label images
- Specification φ
  - φ: Label stop sign as **STOP**
- Check whether there exists an execution of M that violates φ
  - Given a neural network, find if there exists a minor change to a image of stop sign such that M incorrectly classifies?

The Classical Approach

- Given a model M
  - M: A neural network to label images
- Specification φ
  - φ: Label stop sign as **STOP**
- Check whether there exists an execution of M that violates φ
  - Given a neural network, find if there exists a minor change to a image of stop sign such that M incorrectly classifies?
- Yes but so what?

New Challenges

Challenge 1 How do you verify systems that are likely not 100% accurate?

- To err is human after all and AI systems are designed to mimic humans.

(Joint work with Teodora Baluta and Prateek Saxena)

New Challenges

Challenge 1 How do you verify systems that are likely not 100% accurate?

- To err is human after all and AI systems are designed to mimic humans.

(Joint work with Teodora Baluta and Prateek Saxena)

Challenge 2 Probabililstic reasoning is a core component of AI systems?

(Joint work with Sourav Chakraborty – focus of this talk)

From Qualification to Quantification

- The classical verification concerned with finding whether there exists one execution
- The Approach:
  - Represent M and φ as logical formulas and use constraint solver (SAT solvers)
  - Given a formula, a SAT solver checks if there exists a solution
  - $F = (x_1 \vee x_2)$, the SAT solver will return YES

From Qualification to Quantification

- The classical verification concerned with finding whether there exists one execution
- The Approach:
  - Represent M and φ as logical formulas and use constraint solver (SAT solvers)
  - Given a formula, a SAT solver checks if there exists a solution
  - $F = (x_1 \vee x_2)$, the SAT solver will return YES
- We now care whether there exist too many?
  - Given a formula, we need to count
- Challenges: Scalability, encodings, tools, quality of approximations.....

New Challenges

Challenge 1 How do you verify systems that are likely not 100% accurate?

Challenge 2 Probabilistic reasoning is a core component of AI systems? (Joint work with Sourav Chakraborty – focus of this talk)

Probabilistic Reasoning

- Usage of probabilsitic models such as Bayesian networks
- Samplers form the core of the state of the art probabilistic reasoning techniques

Probabilistic Reasoning

- Usage of probabilsitic models such as Bayesian networks
- Samplers form the core of the state of the art probabilistic reasoning techniques
- Usual technique for designing samplers is based on the Markov Chain Monte Carlo (MCMC) methods.

Probabilistic Reasoning

- Usage of probabilsitic models such as Bayesian networks
- Samplers form the core of the state of the art probabilistic reasoning techniques
- Usual technique for designing samplers is based on the Markov Chain Monte Carlo (MCMC) methods.
- Since mixing times/runtime of the underlying Markov Chains are often exponential, several heuristics have been proposed over the years.

Probabilistic Reasoning

- Usage of probabilsitic models such as Bayesian networks
- Samplers form the core of the state of the art probabilistic reasoning techniques
- Usual technique for designing samplers is based on the Markov Chain Monte Carlo (MCMC) methods.
- Since mixing times/runtime of the underlying Markov Chains are often exponential, several heuristics have been proposed over the years.
- Often statistical tests are employed to argue for quality of the output distributions.

Probabilistic Reasoning

- Usage of probabilsitic models such as Bayesian networks
- Samplers form the core of the state of the art probabilistic reasoning techniques
- Usual technique for designing samplers is based on the Markov Chain Monte Carlo (MCMC) methods.
- Since mixing times/runtime of the underlying Markov Chains are often exponential, several heuristics have been proposed over the years.
- Often statistical tests are employed to argue for quality of the output distributions.
- But such statistical tests are often performed on a very small number of samples for which no theoretical guarantees exist for their accuracy.

Uniform Sampler for Discrete Sets

- Implicit representation of a set S: Set of all solutions of φ.
- Given a CNF formula φ, a Sampler $\mathcal{A}$, outputs a random solution of φ.

Definition

A CNF-Sampler, $\mathcal{A}$, is a randomized algorithm that, given a φ, outputs a random element of the set S, such that, for any $\sigma \in S$

$$\Pr[\mathcal{A}(\varphi) = \sigma] = \frac{1}{|S|},$$

Uniform Sampler for Discrete Sets

- Implicit representation of a set S: Set of all solutions of φ.
- Given a CNF formula φ, a Sampler $\mathcal{A}$, outputs a random solution of φ.

Definition

A CNF-Sampler, $\mathcal{A}$, is a randomized algorithm that, given a φ, outputs a random element of the set S, such that, for any $\sigma \in S$

$$\Pr[\mathcal{A}(\varphi) = \sigma] = \frac{1}{|S|},$$

- Uniform sampling has wide range of applications in automated bug discovery, pattern mining, and so on.

Uniform Sampler for Discrete Sets

- Implicit representation of a set S: Set of all solutions of φ.
- Given a CNF formula φ, a Sampler $\mathcal{A}$, outputs a random solution of φ.

Definition

A CNF-Sampler, $\mathcal{A}$, is a randomized algorithm that, given a φ, outputs a random element of the set S, such that, for any $\sigma \in S$

$$\Pr[\mathcal{A}(\varphi) = \sigma] = \frac{1}{|S|},$$

- Uniform sampling has wide range of applications in automated bug discovery, pattern mining, and so on.
- Several samplers available off the shelf: tradeoff between guarantees and runtime

What does Complexity Theory Tell Us

- “far” means total variation distance or the ℓ_1 distance.

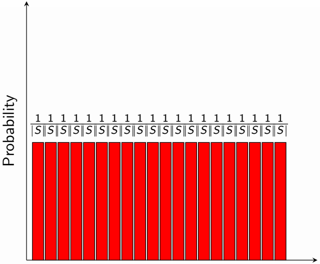


Figure: $\mathcal{U}$: Reference Uniform Sampler

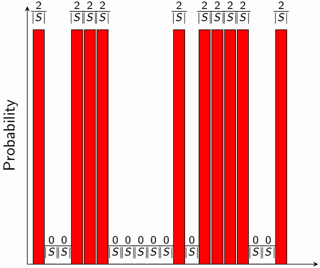


Figure: $\mathcal{A}$: 1/2-far from uniform Sampler

What does Complexity Theory Tell Us

- "far" means total variation distance or the ℓ_1 distance.

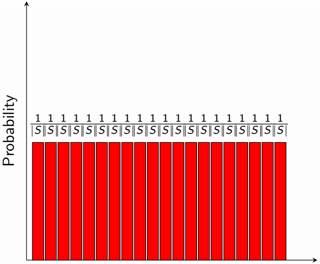


Figure: $\mathcal{U}$: Reference Uniform Sampler

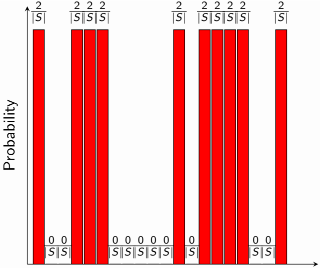


Figure: $\mathcal{A}$: 1/2-far from uniform Sampler

- If $< \sqrt{S}/100$ samples are drawn then with high probability you see only distinct samples from either distribution.

Theorem (Batu-Fortnow-Rubinfeld-Smith-White (JACM 2013))

Testing whether a distribution is ϵ-close to uniform has query complexity $\Theta(\sqrt{|S|}/\epsilon^2)$. [Paninski (Trans. Inf. Theory 2008)]

Beyond Black-Box Testing

Definition (Conditional Sampling)

Given a distribution $\mathcal{D}$ on S one can

- *Specify a set $T \subseteq S$,*
- *Draw samples according to the distribution $\mathcal{D}|_T$, that is, $\mathcal{D}$ under the condition that the samples belong to T.*

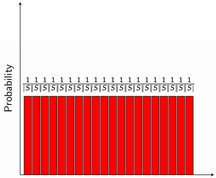
Definition (Conditional Sampling)

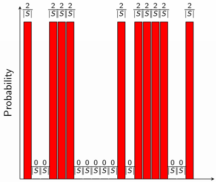
Given a distribution $\mathcal{D}$ on S one can

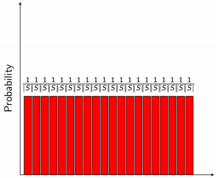
- *Specify a set $T \subseteq S$,*
- *Draw samples according to the distribution $\mathcal{D}|_T$, that is,* *$\mathcal{D}$ under the condition that the samples belong to T.*

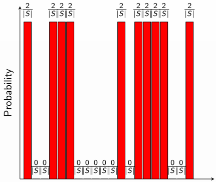
Conditional sampling is at least as powerful as drawing normal samples.
But how more powerful is it?

Testing Uniformity Using Conditional Sampling









An algorithm for testing uniformity using conditional sampling:

1. Draw σ_1 uniformly at random from reference uniform sampler $\mathcal{U}$ and draw σ_2 from sampler under test $\mathcal{A}$. Let $T = \{\sigma_1, \sigma_2\}$.
2. In the case of the "far" distribution, with constant probability, σ_1 will have "low" probability and σ_2 will have "high" probibility.
3. We will be able to distinguish the far distribution from the uniform distribution using constant number of conditional samples from $\mathcal{A}|_T$.
4. The constant depend on the farness parameter.

Input: A sampler under test $\mathcal{A}$, a reference uniform sampler $\mathcal{U}$, a tolerance parameter $\varepsilon > 0$, an intolerance parmaeter $\eta > \varepsilon$, a guarantee parameter δ and a CNF formula φ
Output: ACCEPT or REJECT with the following guarantees:

- if the generator $\mathcal{A}$ is an ε-additive almost-uniform generator then Barbarik ACCEPTS with probability at least $(1 - \delta)$.
- if $\mathcal{A}(\varphi, .)$ is η-far from a uniform generator and If non-adversarial sampler assumption holds then Barbarik REJECTS with probability at least $1 - \delta$.

Theorem

Given ε, η and δ, Barbarik need at most $K = \widetilde{O}(\frac{1}{(\eta-\varepsilon)^4})$ samples for any input formula φ, where the tilde hides a poly logarithmic factor of $1/\delta$ and $1/(\eta - \varepsilon)$.

- $\varepsilon = 0.6, \eta = 0.9, \delta = 0.1$
- Maximum number of required samples $K = 1.72 \times 10^6$
- Independent of the number of variables
- To Accept, we need K samples but rejection can be achieved with lesser number of samples.

Empirical Results

Experimental Setup

- Three state of the art (almost-)uniform samplers
  - UniGen2: Theoretical Guarantees of almost-uniformity
  - SearchTreeSampler: Very weak guarantees
  - QuickSampler: No Guarantees
- Recent study that proposed Quicksampler perform unsound statistical tests and claimed that all the three samplers are indistinguishable

Results-I

Instances	#Solutions	UniGen2		SearchTreeSampler	
		Output	#Samples	Output	#Samples
71	1.14×2^{59}	A	1729750	R	250
blasted_case49	1.00×2^{61}	A	1729750	R	250
blasted_case50	1.00×2^{62}	A	1729750	R	250
scenarios_aig_insertion1	1.06×2^{65}	A	1729750	R	250
scenarios_aig_insertion2	1.06×2^{65}	A	1729750	R	250
36	1.00×2^{72}	A	1729750	R	250
30	1.73×2^{72}	A	1729750	R	250
110	1.09×2^{76}	A	1729750	R	250
scenarios_tree_insert_insert	1.32×2^{76}	A	1729750	R	250
107	1.52×2^{76}	A	1729750	R	250
blasted_case211	1.00×2^{80}	A	1729750	R	250
blasted_case210	1.00×2^{80}	A	1729750	R	250
blasted_case212	1.00×2^{88}	A	1729750	R	250
blasted_case209	1.00×2^{88}	A	1729750	R	250
54	1.15×2^{90}	A	1729750	R	250

Results-II

Instances	#Solutions	UniGen2		QuickSampler	
		Output	#Samples	Output	#Samples
71	1.14×2^{59}	A	1729750	R	250
blasted_case49	1.00×2^{61}	A	1729750	R	250
blasted_case50	1.00×2^{62}	A	1729750	R	250
scenarios_aig_insertion1	1.06×2^{65}	A	1729750	R	250
scenarios_aig_insertion2	1.06×2^{65}	A	1729750	R	250
36	1.00×2^{72}	A	1729750	R	250
30	1.73×2^{72}	A	1729750	R	250
110	1.09×2^{76}	A	1729750	R	250
scenarios_tree_insert_insert	1.32×2^{76}	A	1729750	R	250
107	1.52×2^{76}	A	1729750	R	250
blasted_case211	1.00×2^{80}	A	1729750	R	250
blasted_case210	1.00×2^{80}	A	1729750	R	250
blasted_case212	1.00×2^{88}	A	1729750	R	250
blasted_case209	1.00×2^{88}	A	1729750	R	250
54	1.15×2^{90}	A	1729750	R	250

Take Home Message

- Barbarik can effectively test whether a sampler generates uniform distribution
- Samplers without guarantees, SearchTreeSampler and QuickSampler, fail the uniformity test while sampler with guarantees passes the uniformity test.

Conclusion

- We need methodological approach to verification of AI systems
- Need to go beyond qualitative verification

Conclusion

- We need methodological approach to verification of AI systems
- Need to go beyond qualitative verification
- Sampling is a crucial component of the state of the art probabilistic reasoning systems
- Traditional verification methodology is insufficient

Conclusion

- We need methodological approach to verification of AI systems
- Need to go beyond qualitative verification
- Sampling is a crucial component of the state of the art probabilistic reasoning systems
- Traditional verification methodology is insufficient
- Property testing meets verification: Promise of strong theoretical guarantees with scalability to large instances

Conclusion

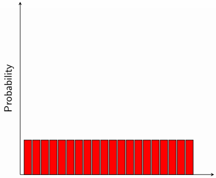
- We need methodological approach to verification of AI systems
- Need to go beyond qualitative verification
- Sampling is a crucial component of the state of the art probabilistic reasoning systems
- Traditional verification methodology is insufficient
- Property testing meets verification: Promise of strong theoretical guarantees with scalability to large instances
- Extend beyond uniform distributions

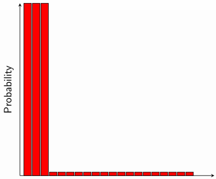
Conclusion

- We need methodological approach to verification of AI systems
- Need to go beyond qualitative verification
- Sampling is a crucial component of the state of the art probabilistic reasoning systems
- Traditional verification methodology is insufficient
- Property testing meets verification: Promise of strong theoretical guarantees with scalability to large instances
- Extend beyond uniform distributions

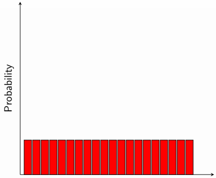
Backup

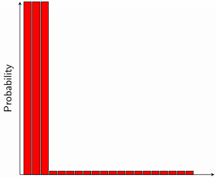
What about other distributions?





What about other distributions?

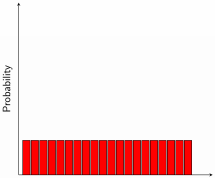


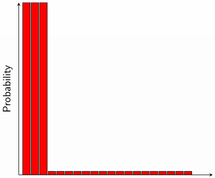


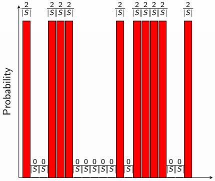
Previous algorithm fails in this case:

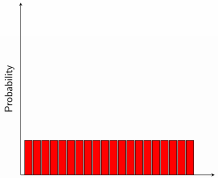
1. Draw two elements σ_1 and σ_2 uniformly at random from the domain. Let $T = \{\sigma_1, \sigma_2\}$.
2. In the case of the "far" distribution, with probability almost 1, both the two elements will have probability same, namely ϵ.
3. Probability that we will be able to distinguish the far distribution from the uniform distribution is very low.

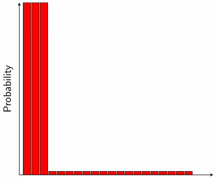
Testing Uniformity Using Conditional Sampling

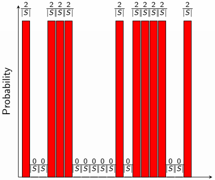












1. Draw σ_1 uniformly at random from the domain and draw σ_2 according to the distribution $\mathcal{D}$. Let $T = \{\sigma_1, \sigma_2\}$.
2. In the case of the "far" distribution, with constant probability, σ_1 will have "low" probability and σ_2 will have "high" probibility.
3. We will be able to distinguish the far distribution from the uniform distribution using constant number of conditional samples from $\mathcal{D}|_T$.
4. The constant depend on the farness parameter.

CNF Samplers

- Input formula: F over variables X
- Challenge: Conditional Sampling over $T = \{\sigma_1, \sigma_2\}$.
- Construct $G = F \wedge (X = \sigma_1 \vee X = \sigma_2)$

CNF Samplers

- Input formula: F over variables X
- Challenge: Conditional Sampling over $T = \{\sigma_1, \sigma_2\}$.
- Construct $G = F \wedge (X = \sigma_1 \vee X = \sigma_2)$
- Most of the samplers enumerate all the points when the number of points in the Domain are small
- Need way to construct formulas whose solution space is large but every solution can be mapped to either σ_1 or σ_2.

Kernel

Input: A Boolean formula φ, two assignments σ_1 and σ_2, and desired number of solutions τ
Output: Formula $\hat{\varphi}$

1. $\tau = |R_{\hat{\varphi}}|$
2. $Supp(\varphi) \subseteq \ Supp(\hat{\varphi})$
3. $z \in R_{\hat{\varphi}} \implies z_{\downarrow S} \in \{\sigma_1, \sigma_2\}$
4. $|\{z \in R_{\hat{\varphi}} \mid z_{\downarrow S} = \sigma_1\}| = |\{z \in R_{\hat{\varphi}} \mid z_{\downarrow S} \cap \sigma_2\}|$, where $S = Supp(\varphi)$.
5. φ and $\hat{\varphi}$ has similar structure

Non-adversarial Sampler

Let $(\hat{\varphi})$ obtained from $kernel(\varphi, \sigma_1, \sigma_2, N)$ such that there are only two set of assignments to variables in φ that can be extended to a satisfying assignment for $\hat{\varphi}$

Definition

The **non-adversarial sampler assumption** states that the distribution of the projection of samples obtained from $\mathcal{A}(\hat{\varphi})$ to variables of φ is same as the conditional distribution of $\mathcal{A}(\varphi)$ restricted to either σ_1 or σ_2

- If $\mathcal{A}$ is a uniform sampler for all the input formulas, it satisfies non-adversarial sampler assumption
- If $\mathcal{A}$ is not a uniform sampler for all the input formulas, it may not necessarily satisfy non-adversarial sampler assumption