PAC-Algorithms for Reliability Estimation of Complex Systems

R. Paredes L. Duenas-Osorio

Department of Civil and Environmental Engineering
Rice University

May 31, 2018

Engineering Mechanics Institute Conference
Boston, MA. USA.

Outline

1. Motivation and Introduction
   - Complex Systems
   - System Reliability
2. PAC Reliability Methods
   - PAC Algorithms
   - Hashing Based Sampling
3. Concrete Application: Network Reliability
   - The problem
   - Relation to Model Counting
   - Computer Experiments
4. Conclusions and Future Work

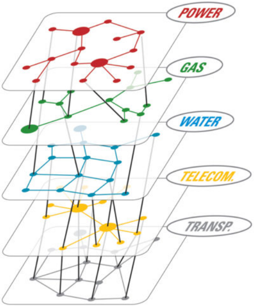
Complex Systems

Feature:

- Interdependencies
- Large scale
- Uncertain
- Chaotic behavior

Critical Infrastructures:

- Power transmission
- Potable water
- Transportation



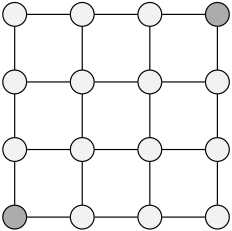
System Reliability

Failure probability of system:

$$p_F = \int_{X \in \Omega} I_F(X) Pr[X] dX$$

Challenges:

- Number of dimensions
- Difficult or no explicit I_F
- Small failure probability



System Reliability - Research Outlook

1. Real System or Surrogate Model
   - First/Second Order Reliability
   - Machine Learning Classifiers (e.g. SVM)
   - Polynomial Chaos Expansion
   - Kriging

2. Design of Experiments
   - Space filling (e.g. Latin Hyper Cube sampling)
   - Variance reduction techniques
   - Model informed collection of samples

3. Estimators with reduced variance
   - Importance Sampling
   - Multilevel Splitting (AKA Subset Simulation)

System Reliability - Challenges

Bottom-line:

- Obtain $\hat{p}_F = \bar{X}_N$
- May know sample variance $\bar{S}_N^2$

... but, what is the error and confidence of our estimate $\hat{p}_F$?

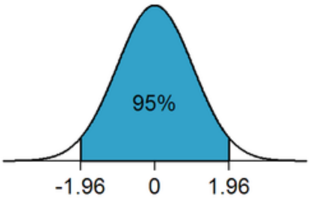
System Reliability - Challenges

Bottom-line:

- Obtain $\hat{p}_F = \bar{X}_N$
- May know sample variance $\bar{S}_N^2$

... but, what is the error and confidence of our estimate $\hat{p}_F$?

Normal assumption (*commonly used*):



Confidence
$1 - \delta = 0.95$
$\hat{p}_F \pm 1.96 \cdot \bar{S}_N / \sqrt{N}$

- Appeals to *CLT* for tail probabilities.
- Uses sample variance, not population variance.

System Reliability - Challenges

Consider $X \sim Bernoulli(p)$.

Trial experiment:

- Sample size $N = 1,000,000$
- Confidence level $1 - \delta = 0.95$
- Experiment replicas $10,000$

Case 1: Fair coin $p = 0.5$

System Reliability - Challenges

Consider $X \sim Bernoulli(p)$.

Trial experiment:

- Sample size $N = 1,000,000$
- Confidence level $1 - \delta = 0.95$
- Experiment replicas $10,000$

Case 1: Fair coin $p = 0.5$

fair coin $p = 0.5$

```
~$ python demo1.py

Confidence 1-delta in practice:

Trial 1: 0.953
Trial 2: 0.950
Trial 3: 0.947 Overconfident!
Trial 4: 0.952
Trial 5: 0.946 Overconfident!
Trial 6: 0.951
Trial 7: 0.948 Overconfident!
Trial 8: 0.948 Overconfident!
Trial 9: 0.954
Trial 10: 0.950 Overconfident!

Aggregating all trials: 0.950
```

System Reliability - Challenges

Biased coin $p = 1 \times 10^{-5}$

```
~$ python demo2.py

Confidence 1-delta in practice:

Trial 1: 0.927 Overconfident!
Trial 2: 0.925 Overconfident!
Trial 3: 0.925 Overconfident!
Trial 4: 0.927 Overconfident!
Trial 5: 0.921 Overconfident!
Trial 6: 0.926 Overconfident!
Trial 7: 0.923 Overconfident!
Trial 8: 0.921 Overconfident!
Trial 9: 0.924 Overconfident!
Trial 10: 0.927 Overconfident!

Agregating all trials: 0.925
```

fair coin $p = 0.5$

```
~$ python demo1.py

Confidence 1-delta in practice:

Trial 1: 0.953
Trial 2: 0.950
Trial 3: 0.947 Overconfident!
Trial 4: 0.952
Trial 5: 0.946 Overconfident!
Trial 6: 0.951
Trial 7: 0.948 Overconfident!
Trial 8: 0.948 Overconfident!
Trial 9: 0.954
Trial 10: 0.950 Overconfident!

Aggregating all trials: 0.950
```

Probably Approximately Correct (PAC) Reliability Methods

Premise: With high probability (probably), return an estimate that does not deviate too much from the exact one (approximately correct)[1].

Formally:

Definition

For input parameters ϵ and δ. A PAC reliability method returns a failure probability estimate $\hat{p}_F$ such that:

$$\Pr[|\hat{p}_F - p_F|/p_F \leq \epsilon] \geq 1 - \delta$$

[1] Borrowed from Lestlie G. Valiant's PAC learning framework (2013)

PAC Reliability Methods - Algorithms

General purpose (ϵ, δ)-approximations that can be used in PAC reliability:

- Optimal Monte Carlo Simulation for bounded random variables (Dagum, Karp, Luby, & Ross, 1995).

$$N \in O(1/p_F \epsilon^{-2} \log 1/\delta)$$

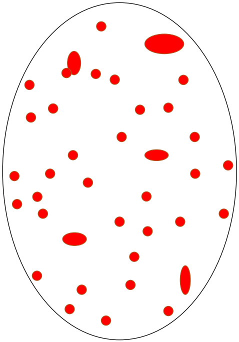
- Estimator for Bernoulli mean with a Gamma Bernoulli Approximation Schemes (Huber, 2017).

$$N \in O(1/p_F \epsilon^{-2} \log 1/\delta)$$

- MCMC based Poisson point process for system reliability and approximation of the mean of Poisson random variables (Walter, 2017; Schott, 2012).

$N \in O(\log(1/p_F)\epsilon^{-2} \log 1/\delta)$, but ignores MCMC mixing time

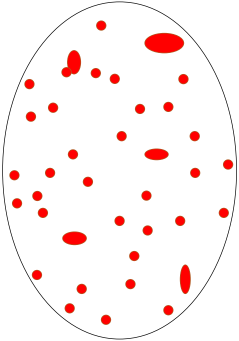
PAC Reliability Methods - Hashing Sampling



$$p_F = \frac{A_R}{A_T}$$

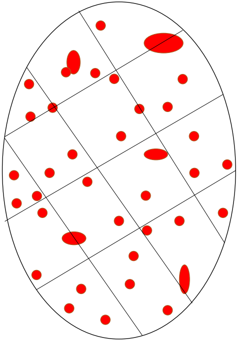
[2](Duenas-Osorio, Meel, Paredes, & Vardi, 2017; Paredes, Duenas-Osorio, Meel, & Vardi, 2018)

PAC Reliability Methods - Hashing Sampling



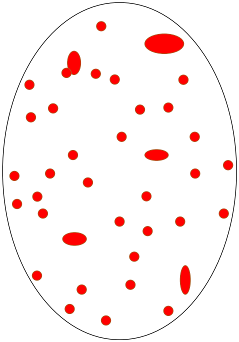
$$p_F = \frac{A_R}{A_T}$$

$$\hat{p}_F = \frac{n_C C_R}{A_T}$$



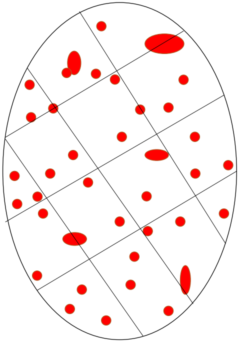
- Break into *roughly equally small* cells (not trivial[2]).
- Solve exact problem for small cell using an *Oracle*.

[2](Duenas-Osorio et al., 2017; Paredes et al., 2018)



$$p_F = \frac{A_R}{A_T}$$

$$\hat{p}_F = \frac{n_C C_R}{A_T}$$



Taking $N \in O(\epsilon^{-2} \log 1/\delta)$ samples delivers: Efficient! (See e.g. FPRAS)

$$\Pr[|\hat{p}_F - p_F|/p_F \leq \epsilon] \geq 1 - \delta$$

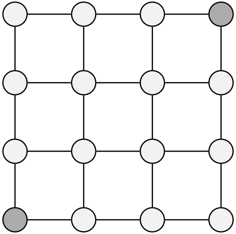
Application to Network Reliability - The Problem

Graph $G(V, E, \mathcal{K})$

V node set

E link set

$\mathcal{K} \subseteq V$ terminal set



Links can fail $X \in \{0, 1\}^{|E|}$

$$I_F(X) = \begin{cases} 1, & \text{if } \mathcal{K} \text{ gets disconnected} \\ 0, & \text{otherwise.} \end{cases}$$

$$p_F = \sum_{X \in \Omega} I_F(X) \Pr[X]$$

Mapping Network Reliability to Model Counting

Define propositional variables $z_u, \forall u \in V$ and $x_{e_i}, \forall e_i \in E$, then:[2]

$$F_{\mathcal{K}} = \exists S(\bigvee_{j \in \mathcal{K}} z_j \wedge \bigvee_{k \in \mathcal{K}} \neg z_k \wedge \bigwedge_{e_i \in E} C_{e_i})$$

with:

$$S = \{z_u | u \in V\}$$

$$C_{e_i} = (z_u \wedge x_{e_i} \rightarrow z_v) \wedge (z_v \wedge x_{e_i} \rightarrow z_u), \forall e_i \in E$$

Equivalence:

$$p_F = \# F_{\mathcal{K}} / 2^{|E|}$$

[2] See details of $\mathcal{K}$-RelNet in our work (Paredes et al., 2018).

Mapping Network Reliability to Model Counting

Define propositional variables $z_u, \forall u \in V$ and $x_{e_i}, \forall e_i \in E$, then:[2]

$$F_{\mathcal{K}} = \exists S (\bigvee_{j \in \mathcal{K}} z_j \wedge \bigvee_{k \in \mathcal{K}} \neg z_k \wedge \bigwedge_{e_i \in E} C_{e_i})$$

with:

$$S = \{z_u | u \in V\}$$

$$C_{e_i} = (z_u \wedge x_{e_i} \rightarrow z_v) \wedge (z_v \wedge x_{e_i} \rightarrow z_u), \forall e_i \in E$$

Equivalence:

$$p_F = \#F_{\mathcal{K}} / 2^{|E|}$$

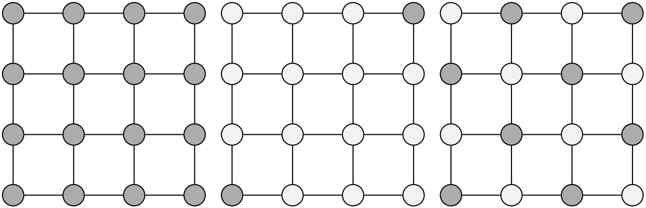
Hashing:

- Efficient random XOR-constraints
- Oracle in practice is a SAT-Solver (handles up to million variables)

[2] See details of $\mathcal{K}$-RelNet in our work (Paredes et al., 2018).

Network Reliability via Hashing Sampling

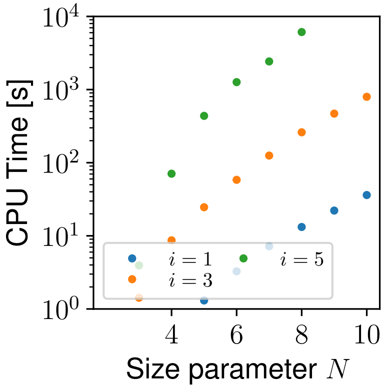
Square grid networks $N \times N$

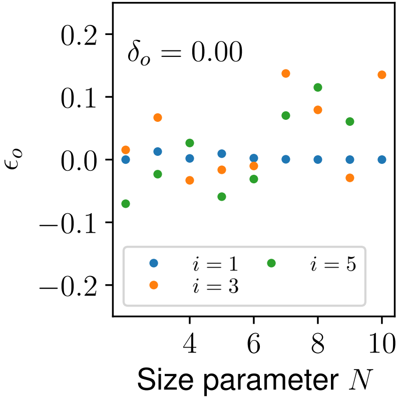


(a) All-Terminal (b) Two-Terminal (c) $\mathcal{K}$-Terminal

Network Reliability via Hashing Sampling

Square grid networks $N \times N$ (edge failure probabilities 2^{-i})

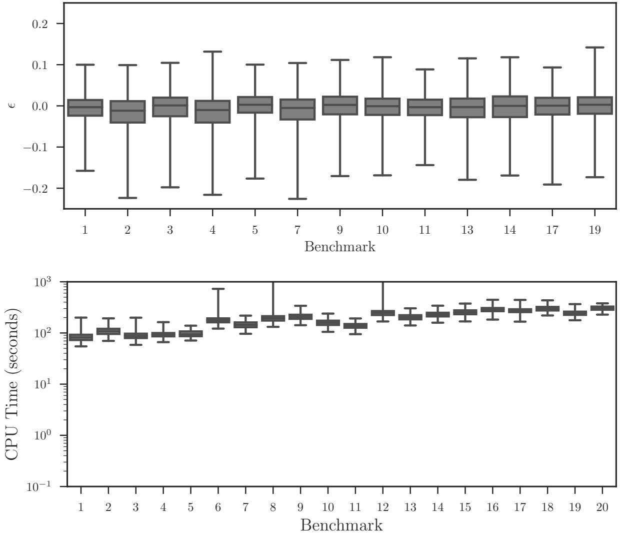




$\mathcal{K}$-RelNet with $\epsilon = 0.8$ and $\delta = 0.2$

Network Reliability via Hashing Sampling

Power Transmission Networks for cities in the U.S.:



Conclusions and Future Work

In contrast to traditional methods, PAC reliability methods:

- deliver proven guarantees of approximation
- empower lay users with input error-confidence parameters
- when efficient, they rigorously mean it!

Conclusions and Future Work

In contrast to traditional methods, PAC reliability methods:

- deliver proven guarantees of approximation
- empower lay users with input error-confidence parameters
- when efficient, they rigorously mean it!

Future work:

- Constrained programming for higher level modeling (e.g. linear and quadratic programming)
- Hybrid domains for handling both continuous and discrete systems
- Improve guarantees of approximation, currently somewhat conservative

Acknowledgements

Joint work with:

K.S. Meel, Computer Science Department, National University of Singapore, Singapore.

M.Y. Vardi, Department of Computer Science, Rice University, Houston, Texas, USA

email: roger.paredes@rice.edu

URL (slides and pertinent papers): github.com/paredesroger/emi2018

Dagum, P., Karp, R., Luby, M., & Ross, S. (1995, jan). An optimal algorithm for Monte Carlo estimation. In *Proceedings of ieee 36th annual foundations of computer science* (Vol. 29, pp. 142–149). IEEE Comput. Soc. Press.

Duenas-Osorio, L., Meel, K. S., Paredes, R., & Vardi, M. Y. (2017). Counting-based Reliability Estimation for Power-Transmission Grids. In *Proceedings of aaai conference on artificial intelligence.* San Francisco.

Paredes, R., Duenas-Osorio, L., Meel, K. S., & Vardi, M. Y. (2018). Network Reliability Estimation in Theory and Practice. *Preprint submitted to: Reliability Engineering & System Safety*.

Schott, S. (2012). *TPA : A new method for approximate counting* (PhD thesis). Duke University.

Walter, C. (2017, jan). Point process-based Monte Carlo estimation. *Statistics and Computing*, *27*(1), 219–236.