Reachability analysis of Markov models via generating functions

Arpit Sharma, Michele Boreale

UNIVERSITÀ DEGLI STUDI FIRENZE
DiSIA
DIPARTIMENTO DI STATISTICA, INFORMATICA, APPLICAZIONI "GIUSEPPE PARENTI"

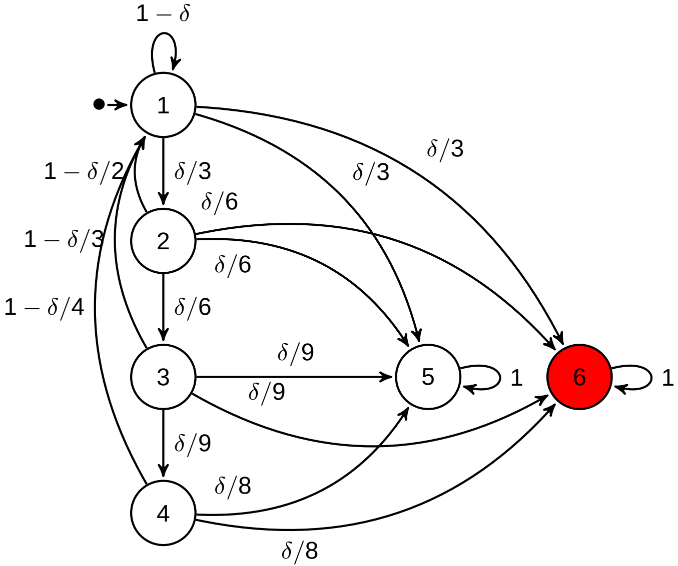
January 20, 2016

Why Markov models?

Markov chains are a very natural model:

- communication systems and protocols
- embedded systems
- page ranking (Google)
- systems biology
- classical performance and dependability analysis

Discrete-time Markov chain



Reachability Computation (Old)

- Consider a DTMC $X_0, X_1, X_2, \ldots$ over $1..N$, with $X_0 = 1$.
  Focus on **reachability**

$$p_{reach} = \Pr(X_j = N \text{ for some } j)$$

Reachability Computation (Old)

- Consider a DTMC $X_0, X_1, X_2, \ldots$ over 1..N, with $X_0 = 1$. Focus on **reachability**

$$p_{reach} = \Pr(X_j = N \text{ for some } j)$$

- In the traditional way, solve a (large) linear system. $p_{reach} = y_1$ with (column-wise)

$$y_i = \sum_{j \neq N} P(j,i) y_j + P(N,i)$$

- Whole state space stored : $O(N^2)$ memory, $O(N^3)$ time.

Problem : State Space Grows Exponentially

Can we do better?

Is it possible to

- compute on-the-fly with only black-box access to transition function
- not store the entire state space
- possibly approximate, but with very accurate results

Generating Functions [Boreale, ICALP 2015]

Moments: $a_0, a_1, a_2, ...$ where
$a_j = \Pr(X_j = N \text{ and } X_i \neq N \text{ for } i < j)$
Generating function

$$g(z) \quad = \quad a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \cdots$$

Why generating functions?

Generating Functions [Boreale, ICALP 2015]

Moments: $a_0, a_1, a_2, ...$ where
$a_j = \Pr(X_j = N \text{ and } X_i \neq N \text{ for } i < j)$
Generating function

$$g(z) \;=\; a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \cdots$$

Why generating functions?

$$p_{reach} \;=\; g(1)$$

Generating Functions [Boreale, ICALP 2015]

Moments: $a_0, a_1, a_2, ...$ where
$a_j = \Pr(X_j = N \text{ and } X_i \neq N \text{ for } i < j)$
Generating function

$$\begin{aligned} g(z) &= a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \cdots \\ g'(z) &= 1 \cdot a_1 + 2 \cdot a_2 z + 3 \cdot a_3 z^2 + \cdots \end{aligned}$$

Why generating functions?
Let **hitting time** $T = \min\{j : X_j = N\}$

$$\begin{aligned} p_{reach} &= g(1) \\ E[T|Reach] &= g'(1)/g(1) \end{aligned}$$

Generating Functions [Boreale, ICALP 2015]

Moments: $a_0, a_1, a_2, ...$ where
$a_j = \Pr(X_j = N \text{ and } X_i \neq N \text{ for } i < j)$
Generating function

$$\begin{aligned} g(z) &= a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \cdots \\ g'(z) &= 1 \cdot a_1 + 2 \cdot a_2 z + 3 \cdot a_3 z^2 + \cdots \\ g''(z) &= 2 \cdot a_2 + 6 \cdot a_3 z + \cdots \end{aligned}$$

Why generating functions?
Let **hitting time** $T = \min\{j : X_j = N\}$

$$\begin{aligned} p_{reach} &= g(1) \\ E[T|Reach] &= g'(1)/g(1) \\ Var[T|Reach] &= \left(g''(1) + g'(1)\right)/g(1) - \left(g'(1)/g(1)\right)^2 \\ &\cdots \end{aligned}$$

With $g(z)$, a whole host of information about the system is packed into a single mathematical object! **How to extract this information**?

Rationality of $g(z)$

- Moments can be computed "on the fly"
- $a_{j+1} = e_N^T P^{j+1} e_1 - e_N^T P^j e_1 = e_N^T P^j (P - I) e_1 = e_N^T P^j \tilde{e}_1$
- Key observation : $g(z)$ is **rational**: $g(z) = r(z)/t(z)$
- $g(z) \quad = \quad a_0 \quad + \quad z \cdot e_N^T \mathbf{(I - zP)^{-1}} \tilde{e}_1$

Solution Approach 1

- compute enough moments of $g(z)$ on the fly, $a_0, ..., a_{2N-1}$
- reconstruct $g(z)$ exactly from those $2N$ moments, using Pade approximants theory.
- this takes $O(N)$ memory, but it is costly and unstable numerically for large N.

Solution 2 - Building $\hat{g}(z)$ via Arnoldi

- choose a small m-dimensional subspace, $\mathcal{K}_m \subseteq \mathbb{R}^N$ $(m \ll N)$
- **project** P onto $\mathcal{K}_m$, thus obtaining a low-dimensional system

$$H_m = V_m^T P V_m$$

- build the g.f. $\hat{g}(z)$ of the small system in matrix form

$$\hat{g}(z) \quad = \quad a_0 \quad + \quad z \cdot (e_N^T V_m)\,\mathbf{(I - zH_m)^{-1}}\,(V_m^T \tilde{e}_1)$$

Solution 2 - Building $\hat{g}(z)$ via Arnoldi

- choose a small m-dimensional subspace, $\mathcal{K}_m \subseteq \mathbb{R}^N$ ($m \ll N$)
- **project** P onto $\mathcal{K}_m$, thus obtaining a low-dimensional system

$$H_m = V_m^T P V_m$$

- build the g.f. $\hat{g}(z)$ of the small system in matrix form

$$\hat{g}(z) = a_0 + z \cdot (e_N^T V_m) \, (\mathbf{I} - \mathbf{z}\mathbf{H}_\mathbf{m})^{-\mathbf{1}} \, (V_m^T \tilde{e}_1)$$

$\mathcal{K}_m = \mathrm{span}\{\tilde{e}_1, P\tilde{e}_1, ..., P^{m-1}\tilde{e}_1\}$ (Krylov space)

Solution 2 - Building $\hat{g}(z)$ via Arnoldi

- choose a small m-dimensional subspace, $\mathcal{K}_m \subseteq \mathbb{R}^N$ ($m \ll N$)
- **project** P onto $\mathcal{K}_m$, thus obtaining a low-dimensional system

$$H_m = V_m^T P V_m$$

- build the g.f. $\hat{g}(z)$ of the small system in matrix form

$$\hat{g}(z) \quad = \quad a_0 \quad + \quad z \cdot (e_N^T V_m)\,\mathbf{(I - zH_m)^{-1}}\,(V_m^T \tilde{e}_1)$$

$\mathcal{K}_m = \mathrm{span}\{\tilde{e}_1, P\tilde{e}_1, ..., P^{m-1}\tilde{e}_1\}$ (Krylov space)

Arnoldi algo - Efficient way to compute H_m and V_m

Experiments

System			Proposed method				Prism			
Name	N	p_{reach}	$\hat{p}_{reach}$	%*error*	m	*time* (s)	$\hat{p}_{reach}$	%*error*	m	*time* (s)
Nasty	10^5	0.5	0.5000	$< \mathbf{10^{-11}}$	5	**0.08**	0.4999	$< 10^{-4}$	27	5524
	10^6	0.5	0.5000	$< \mathbf{10^{-10}}$	5	**0.06**	-	-	-	-
Queue	49	0.25	0.2500	$< \mathbf{10^{-5}}$	15	0.92	0.2454	1.81	133176	**0.34**
	231	0.125	0.1248	**0.11**	37	**11.21**	0.0013	98.91	1983765	22.63
Ising	64	0.5	0.5000	$< \mathbf{10^{-3}}$	9	2.06	0.4982	0.34	26433	**0.05**
	256	0.5	0.4998	$<$**0.10**	18	20.58	0.0578	88.42	1257073	**8.16**
Chemical	25	0.5	0.4994	**0.10**	18	**0.40**	1.2×10^{-7}	99.99	2000030	1.55
	27	0.5	0.4937	**1.25**	20	**0.45**	7.8×10^{-9}	99.99	2000034	1.96

Inspired by this

We are investigating if this technique could be used for efficient reachability analysis of :

1. discrete-time Markov decision processes (DMDPs)
2. continuous-time Markov chains (CTMCs)

MDP Analysis

- Goal : compute maximum probability to reach N
- $C^*(i)$ = max Pr(reaching N starting from state i), where max is taken over all schedulers
- in fact, mostly interested in 1 as a starting state, hence in $C^*(1)$

Approach 1 - Accelerating value iteration with Pade

- compute vector C_{j+1}, where $C_{j+1}(i) = max_a \sum_{s'} P_a(s', i) \cdot C_j(i)$, i.e., maximum probability to reach N from i within $j+1$ steps, hence
- we approximate max probability of reaching N under optimal scheduler in $j+1$ steps with C_{j+1}
- compute $C_0(1)$,$C_1(1)$,$C_2(1)$,$C_3(1)$
- compute moments $a_{j+1} = C_{j+1}(1) - C_j(1)$
- compute $\hat{g}(z)$ by P.A. of first m moments
- problem : slow, unstable numerically and does not compare well with PRISM model checker

Approach 2 - Policy iteration

1. Policy, π : *States* $\rightarrow$ *Act*
2. Select random initial policy π_0, and repeat until convergence
   2.1 evaluate current policy π_j using Arnoldi-like algorithm on the resulting *MC*
   2.2 update policy π_j according to the equation $\pi_{j+1}(i) = max_a \sum_{s'} P_a(s', i) \cdot C_j(i)$
3. steps 2.1, 2.2 can be still performed on the fly, with suitable data structures

Future work - CTMC Analysis

- get $g(z)$ as Laplace transform of Chapman-Kolmogoroff differential equation
- apply Arnoldi algorithm for computation of $\hat{g}(z)$

References

- A.C. Antoulas. *Approximation of Large-scale Dynamical Systems*. SIAM, 2005.
- W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, vol. 9, pp. 17-29, 1951.
- G. Baker Jr. *Essentials of Pad√©Approximants.AcademicPress*, 1975.
- *M. Boreale. Full version of the present paper, Matlab and* PRISM *code.* `http://rap.dsi.unifi.it/~boreale/papers/GFviaKrylov.rar`
- *P. Doyle. Markov chains via generating functions. Manuscript, 2010.* `https://math.dartmouth.edu/~doyle/docs/mc/mc.ps`
- *D.J. Hartfiel, Carl D. Meyer. On the structure of stochastic matrices with a subdominant eigenvalue near 1. Linear Algebra Appl. 272, pp. 193-203, 1998.*
- *B. Philippe, Y. Saad , W.J. Stewart. Numerical Methods in Markov Chain Modelling. Operations Research, vol. 40, pp. 1156–1179, 1996.*
- *Y. Saad. Iterative methods for sparse linear systems. SIAM, 2003.*
- *H.S. Wilf. Generatingfunctionology, 2/e. Academic Press, 1994.*

Thank You