Multi-variable Markov Process: a matrix product states treatment

Collaboration:

Bin Xu – Physics
Peiqi Wang – ORFE
Liangsheng Zhang – Physics
Peter Gjeltema – MAE
Jun Xiong – Physics

CONTENTS

- Transitional matrix method of Markov process
- Matrix product states
- The Algorithm and Program structures
- Some results

MARKOV PROCESS

- An emotional boy: happy or angry
- IF he is happy:
  - 50% -> Angry
  - 50% -> Remain happy
- IF he is angry:
  - 50% -> Remain angry
  - 50% -> Happy

Transitional Matrix

$$v_{t+1} = T \cdot v_t$$

$$\begin{pmatrix} 0.5P_{Angry} + 0.5P_{Happy} \\ 0.5P_{Angry} + 0.5P_{Happy} \end{pmatrix} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix} \begin{pmatrix} P_{Angry} \\ P_{Happy} \end{pmatrix}$$

MARKOV PROCESS

- 2 emotional boys
- With probability p, they will change their states TOGETHER
- Otherwise, they remain in their states

Transitional Matrix

$$\begin{pmatrix} 1-p & 0 & 0 & p \\ 0 & 1-p & p & 0 \\ 0 & p & 1-p & 0 \\ p & 0 & 0 & 1-p \end{pmatrix} \begin{pmatrix} P_{AA} \\ P_{AH} \\ P_{HA} \\ P_{HH} \end{pmatrix}$$

$$T = (1-p)I + p\sigma^X \otimes \sigma^X$$

$$\sigma^X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

MARKOV PROCESS

- N emotional boys
- With probability p, a pair nearest neighbors will change states TOGETHER
- Otherwise, they remain their states

Transitional Matrix

$$T = pI + (1-p)\sum_{i=1}^{n-1} \frac{1}{n-1} \sigma_i^x \otimes \sigma_{i+1}^x$$

$$\sigma_i^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

MARKOV PROCESS

- N emotional boys
- With probability p, a pair nearest neighbors will change states TOGETHER
- Otherwise, they remain their states

Transitional Matrix

$$T = pI + (1-p)\sum_{i=1}^{n-1} \frac{1}{n-1} \sigma_i^x \otimes \sigma_{i+1}^x$$

$$\sigma_i^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

The T matrix is 2^n dimensional!

MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

2^N states

MARKOV PROCESS

- How do we describe a state of N boys?

- Read as a function

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

2^N states

MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

2^N states

- Read as a function

$$P : \{A, H\}^{\otimes N} \to \mathbb{R}$$

MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

2^N states

- Read as a function

$$P : \{A, H\}^{\otimes N} \to \mathbb{R}$$

- How do we represent a function?

MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

2^N states

- Read as a function

$$P : \{A, H\}^{\otimes N} \to \mathbb{R}$$

- How do we represent a function?
  - Table of value

MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

2^N states

- Read as a function

$$P : \{A, H\}^{\otimes N} \to \mathbb{R}$$

- How do we represent a function?
  - Table of value
  - Analytical relations

MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

2^N states

- Read as a function

$$P : \{A, H\}^{\otimes N} \to \mathbb{R}$$

- How do we represent a function?
  - Table of value
  - Analytical relations
    - (Taylor expansion, etc.)

MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

2^N states

- Read as a function

$$P : \{A, H\}^{\otimes N} \to \mathbb{R}$$

- How do we represent a function?
  - Table of value
  - Analytical relations
    - (Taylor expansion, etc.)
  - Algebraic relations

MARKOV PROCESS

- How do we describe a state of N boys?

$$\begin{pmatrix} P(AAA\cdots AA) \\ P(AAA\cdots AH) \\ P(AAA\cdots HA) \\ P(AAA\cdots HH) \\ \cdots \\ P(HHH\cdots HH) \end{pmatrix}$$

2^N states

- Read as a function

$$P : \{A, H\}^{\otimes N} \to \mathbb{R}$$

- How do we represent a function?
  - Table of value
  - Analytical relations
    - (Taylor expansion, etc.)
  - Algebraic relations
    - (product of matrices)

MATRIX PRODUCT STATES (MPS)

- Algebraic representation of the "state function"

$$P : \{A, H\}^{\otimes N} \to \mathbb{R}$$
$$\sigma_1\sigma_2\cdots\sigma_N \mapsto M^{[1]}_{\sigma_1} M^{[2]}_{\sigma_2} M^{[3]}_{\sigma_3} \cdots M^{[N]}_{\sigma_N}$$

- We need to store 2N matrices instead of 2^N numbers

- for example:

$$P(AHA\cdots HA) = M^{[1]}_A M^{[2]}_H M^{[3]}_A \cdots M^{[N-1]}_H M^{[N]}_A$$

- In the memory

$M^{[1]}_A$	$M^{[2]}_A$	$\cdots$	$M^{[N]}_A$
$M^{[1]}_H$	$M^{[2]}_H$	$\cdots$	$M^{[N]}_H$

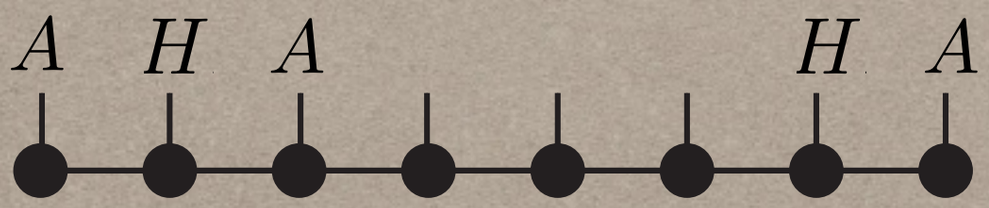
MATRIX PRODUCT STATES (MPS)

- Algebraic representation of the "state function"

$$P : \{A, H\}^{\otimes N} \to \mathbb{R}$$
$$\sigma_1\sigma_2\cdots\sigma_N \mapsto M^{[1]}_{\sigma_1} M^{[2]}_{\sigma_2} M^{[3]}_{\sigma_3} \cdots M^{[N]}_{\sigma_N}$$

- Pictorial representation

$$P(AHA\cdots HA) = M^{[1]}_A M^{[2]}_H M^{[3]}_A \cdots M^{[N-1]}_H M^{[N]}_A$$



- In the memory

$M^{[1]}_A$	$M^{[2]}_A$	$\cdots$	$M^{[N]}_A$
$M^{[1]}_H$	$M^{[2]}_H$	$\cdots$	$M^{[N]}_H$

MATRIX PRODUCT STATES (MPS)

MATRIX PRODUCT STATES (MPS)

- So we solve an exponentially hard problem in polynomial time!

MATRIX PRODUCT STATES (MPS)

- So we solve an exponentially hard problem in polynomial time!
- “Really? That sounds too good to be true!”

MATRIX PRODUCT STATES (MPS)

- So we solve an exponentially hard problem in polynomial time!
- “Really? That sounds too good to be true!”
- I cheated a little bit: the dimension of these matrices (bound dimension) will be in general 2^N

MATRIX PRODUCT STATES (MPS)

- So we solve an exponentially hard problem in polynomial time!
- “Really? That sounds too good to be true!”
- I cheated a little bit: the dimension of these matrices (bound dimension) will be in general 2^N
- But for many “interesting” problems, a rather small matrix approximates the state very nicely!

MATRIX PRODUCT STATES (MPS)

- So we solve an exponentially hard problem in polynomial time!
- “Really? That sounds too good to be true!”
- I cheated a little bit: the dimension of these matrices (bound dimension) will be in general 2^N
- But for many “interesting” problems, a rather small matrix approximates the state very nicely!
- We are just exploiting the hidden structure of the model!

MATRIX PRODUCT STATES (MPS)

- So we solve an exponentially hard problem in polynomial time!
- “Really? That sounds too good to be true!”
- I cheated a little bit: the dimension of these matrices (bound dimension) will be in general 2^N
- But for many “interesting” problems, a rather small matrix approximates the state very nicely!
- We are just exploiting the hidden structure of the model!
    - Remember image compression using Fourier analysis or wavelet? This is the same logic!

MATRIX PRODUCT OPERATORS (MPO)

- The transitional matrix is a product of matrices, where each matrix operates on a single site.

$$T = pI + (1-p)\sum_{i=1}^{n-1} \frac{1}{n-1}\sigma_i^x \otimes \sigma_{i+1}^x$$

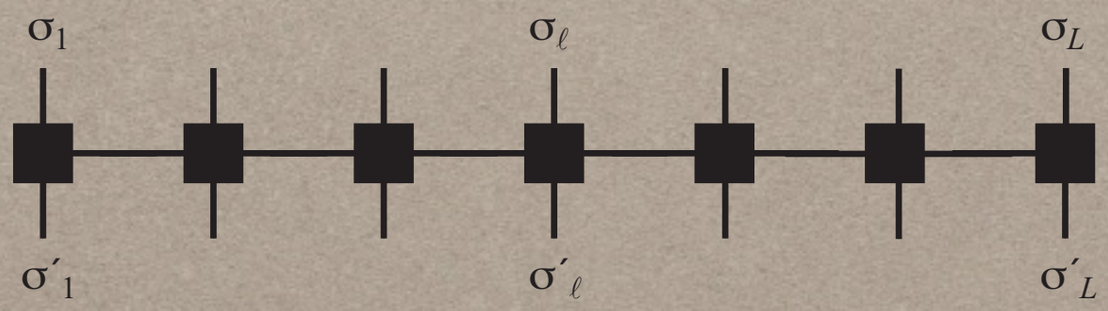
$$T = T^{[1]}T^{[2]} \ldots T^{[N]}$$

$$T^{[1]} = \begin{bmatrix} pI & (1-p)\sigma_1^X & I \end{bmatrix}$$

$$T^{[i]} \begin{bmatrix} I & 0 & 0 \\ \sigma^X & 0 & 0 \\ 0 & (1-p)\sigma_i^X & I \end{bmatrix}$$

$$T^{[N]} = \begin{bmatrix} I \\ \sigma_N^X \\ 0 \end{bmatrix}$$

- Pictorial representation



MATRIX PRODUCT OPERATORS (MPO)

- The transitional matrix is a product of matrices, where each matrix operates on a single site.

$$T = pI + (1-p)\sum_{i=1}^{n-1} \frac{1}{n-1}\sigma_i^x \otimes \sigma_{i+1}^x$$

$$T = T^{[1]}T^{[2]}\ldots T^{[N]}$$

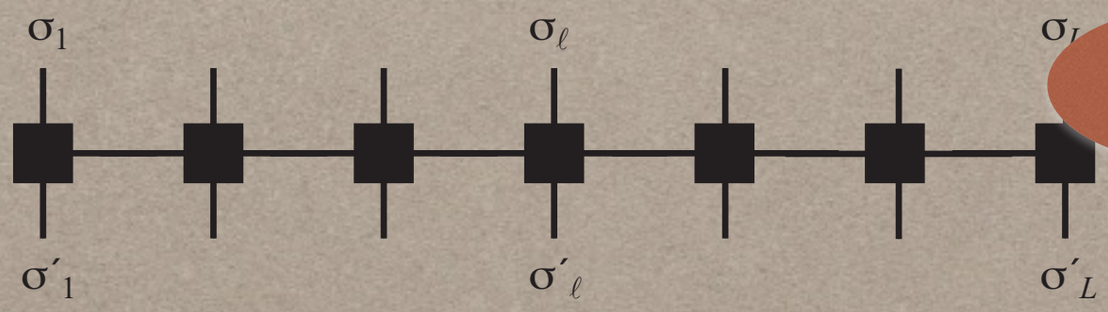
first

$$T^{[1]} = \begin{bmatrix} pI & (1-p)\sigma_1^X & I \end{bmatrix}$$

middle

$$T^{[i]} \begin{bmatrix} I & 0 & 0 \\ \sigma^X & 0 & 0 \\ 0 & (1-p)\sigma_i^X & I \end{bmatrix}$$

- Pictorial representation



last

$$T^{[N]} = \begin{bmatrix} I \\ \sigma_N^X \\ 0 \end{bmatrix}$$

MATRIX PRODUCT OPERATORS (MPO)

- The transitional matrix is a product of matrices, where each matrix operates on a single site.

$$T = pI + (1-p)\sum_{i=1}^{n-1} \frac{1}{n-1}\sigma_i^x \otimes \sigma_{i+1}^x$$

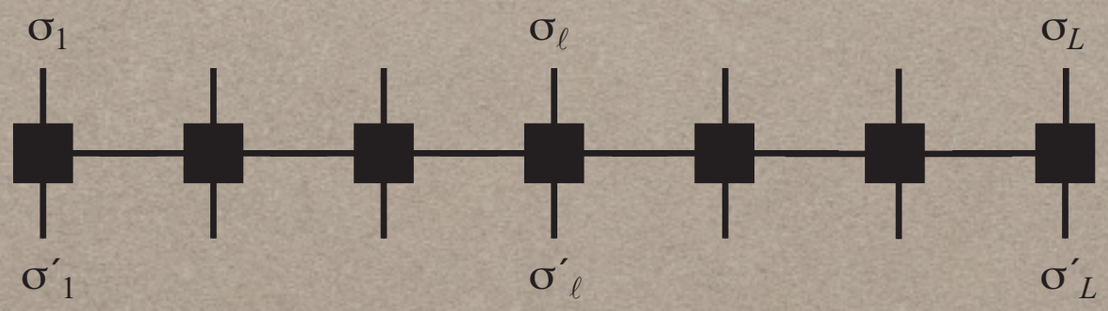
$$T = T^{[1]}T^{[2]} \ldots T^{[N]}$$

$$T^{[1]} = \begin{bmatrix} pI & (1-p)\sigma_1^X & I \end{bmatrix}$$

$$T^{[i]} \begin{bmatrix} I & 0 & 0 \\ \sigma^X & 0 & 0 \\ 0 & (1-p)\sigma_i^X & I \end{bmatrix}$$

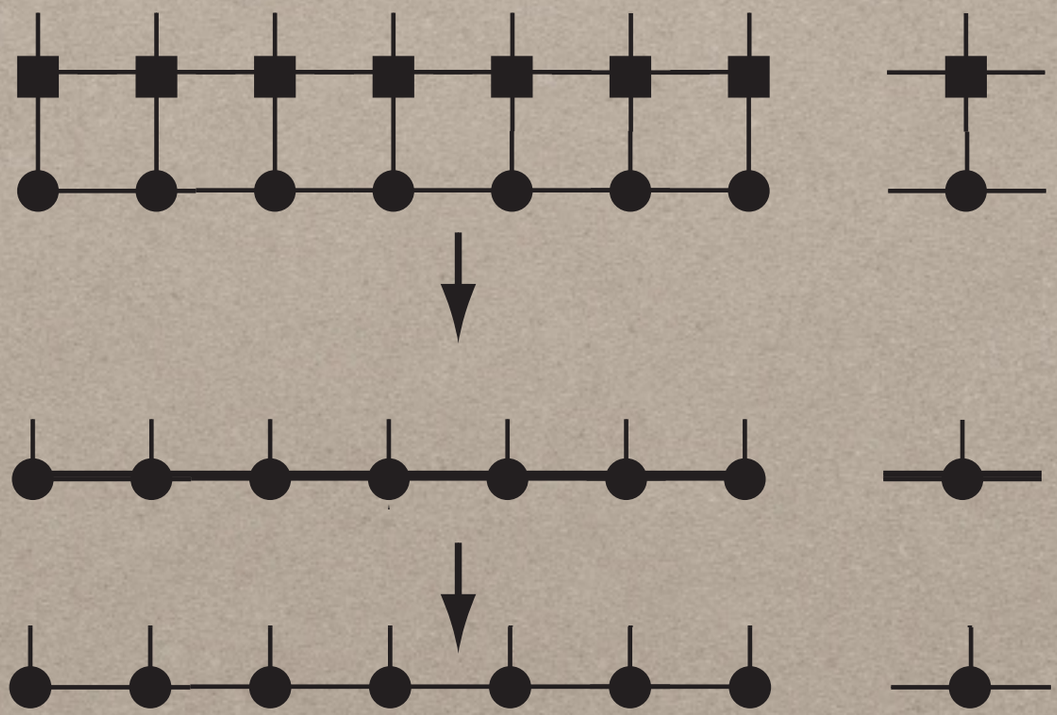
$$T^{[N]} = \begin{bmatrix} I \\ \sigma_N^X \\ 0 \end{bmatrix}$$

- Pictorial representation



MATRIX PRODUCT OPERATORS (MPO)

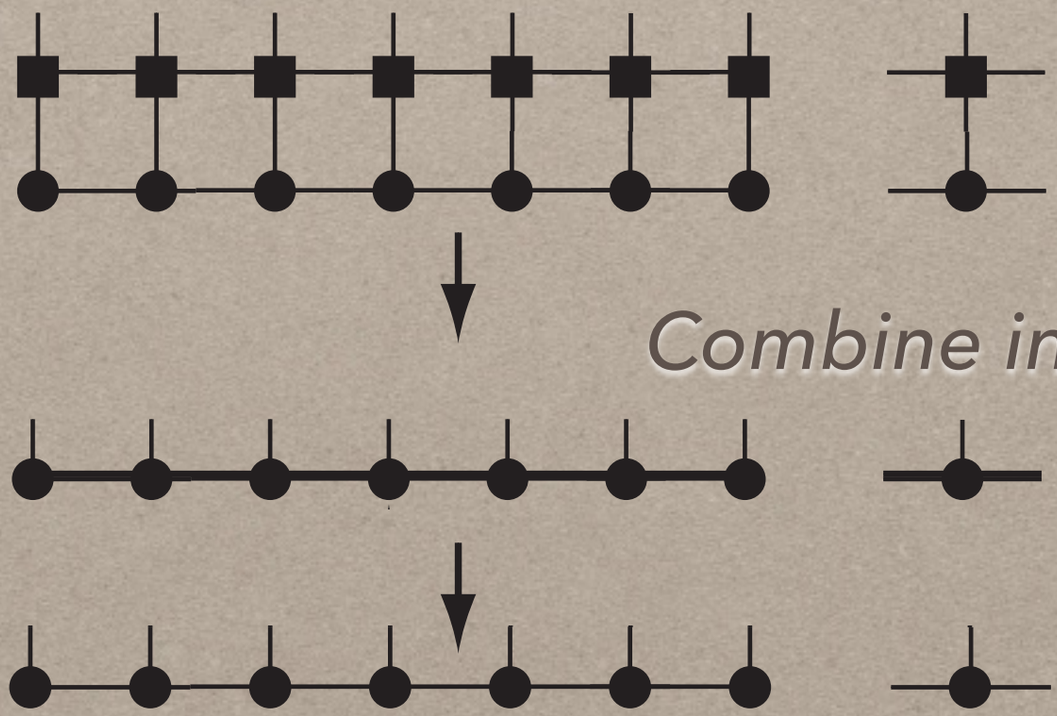
- Applying MPO of the transitional matrix on the MPS of a state vector



$$v_{t+1} = T \cdot v_t$$

MATRIX PRODUCT OPERATORS (MPO)

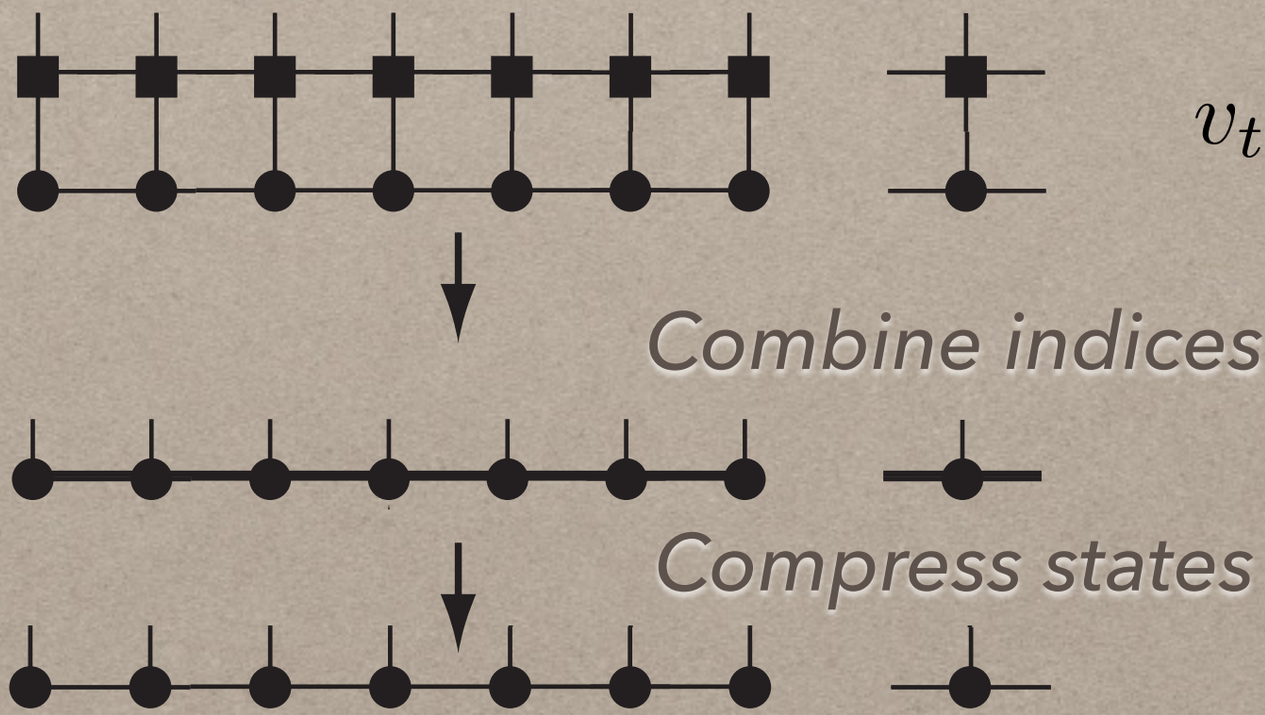
- Applying MPO of the transitional matrix on the MPS of a state vector



$$v_{t+1} = T \cdot v_t$$

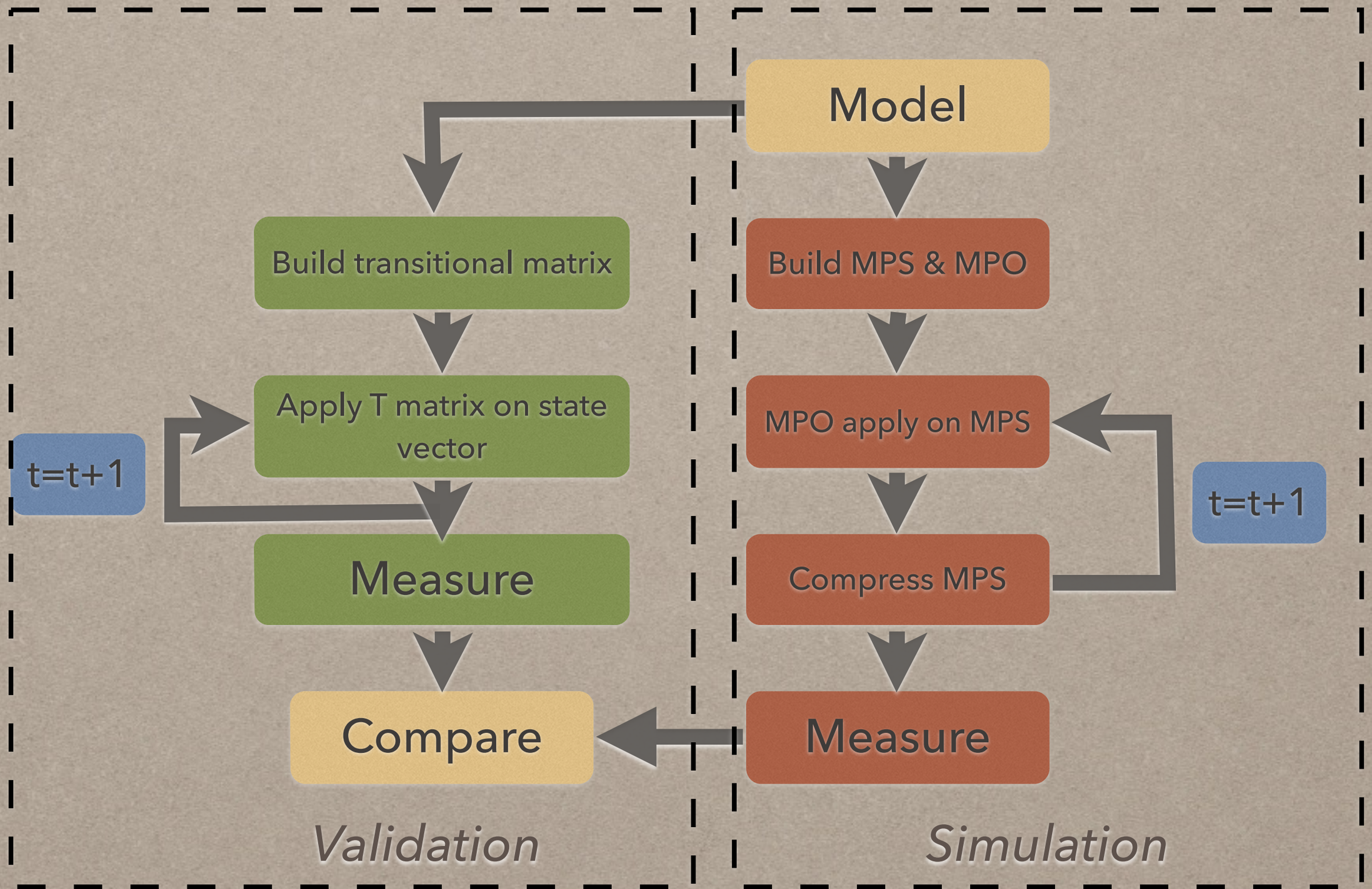
MATRIX PRODUCT OPERATORS (MPO)

- Applying MPO of the transitional matrix on the MPS of a state vector

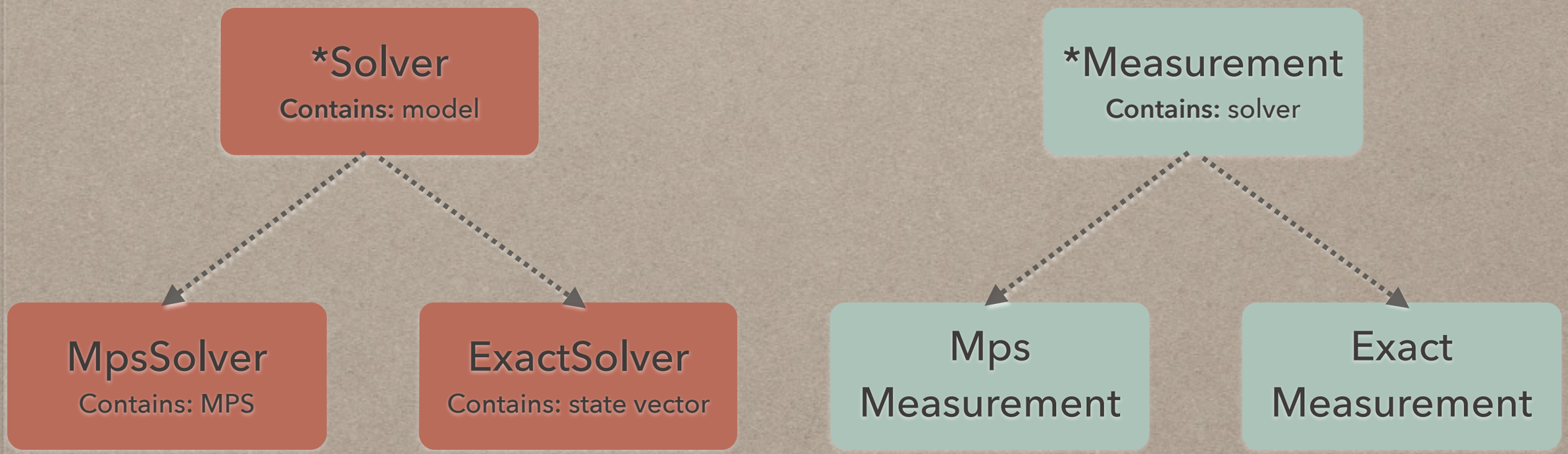


$$v_{t+1} = T \cdot v_t$$

ALGORITHM


Model
Build transitional matrix
Build MPS & MPO
Apply T matrix on state vector
MPO apply on MPS
t=t+1
t=t+1
Measure
Compress MPS
Compare
Measure
Validation
Simulation

CLASS STRUCTURE



THE PROGRAM

binarybin / **MPS_Proba**

Unwatch 5 | Unstar 4 | Fork 1

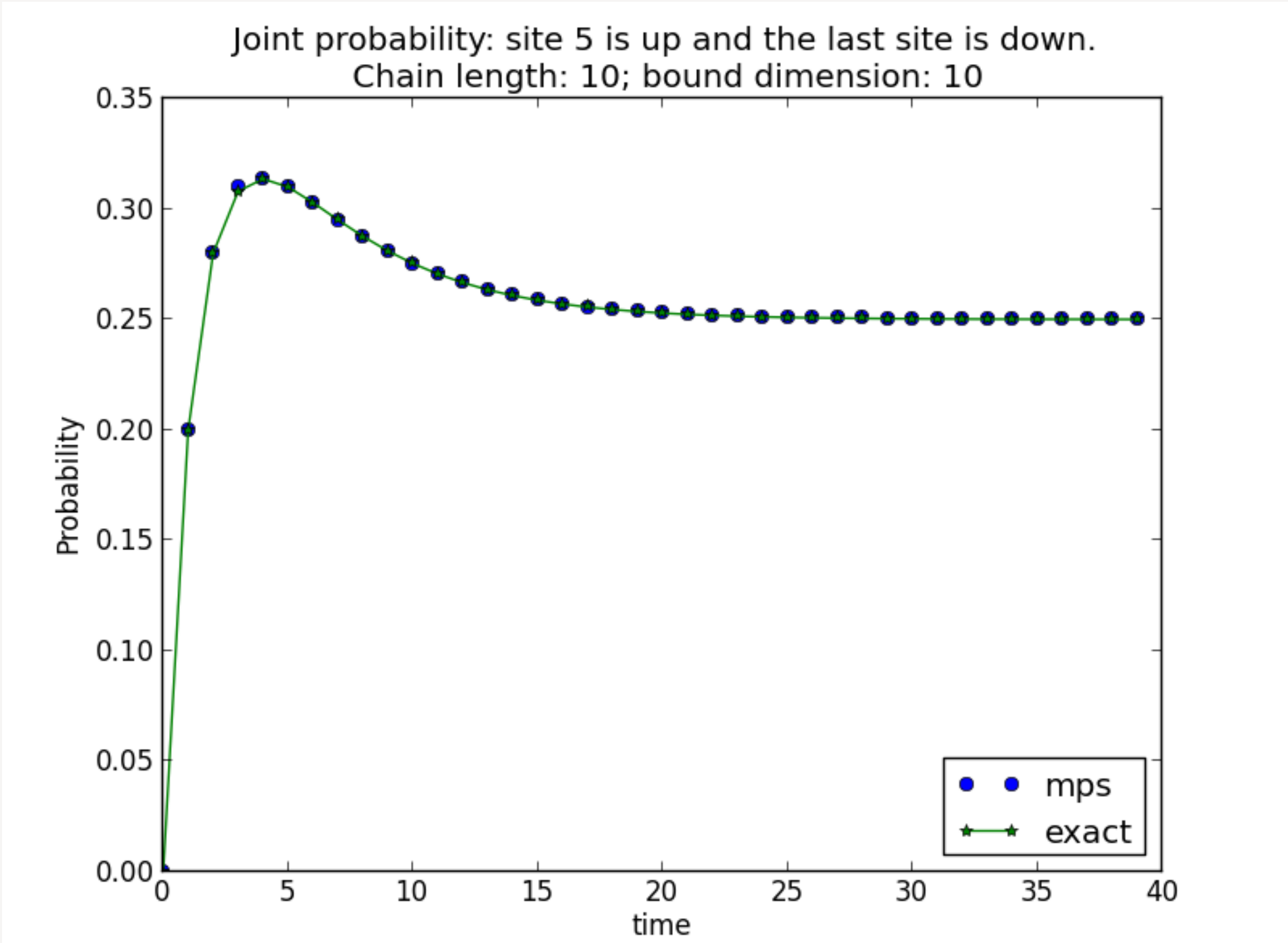
branch: **master** | **MPS_Proba** / **src** / +

variance measurement bug

binarybin authored 16 hours ago — latest commit `172f10aedb`

..		
angryboys.py	improved model representation (for all models)	14 days ago
exactmeasurement.py	variance measurement bug	16 hours ago
exactsolver.py	Abolished output channels; build MPS and exact representations of mod…	14 days ago
exponentialboys.py	improved model representation (for all models)	14 days ago
measurement.py	debugged exact measurement	16 hours ago
model.py	improved model representation (for all models)	14 days ago
mpsmeasurement.py	debugged exact measurement	16 hours ago
mpssolver.py	ProjectionBoys model added: involving pi_plus and pi_minus projection…	14 days ago
proba.py	debugged exact measurement	16 hours ago
projectionboys.py	improved model representation (for all models)	14 days ago
radiatingboys.py	fixed a bug in radiatingboys model	14 days ago
solver.py	ProjectionBoys model added: involving pi_plus and pi_minus projection…	14 days ago

RESULTS


Joint probability: site 5 is up and the last site is down.
Chain length: 10; bound dimension: 10
Probability
time
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
5
10
15
20
25
30
35
40
mps
exact