

# Solving linear equation systems with Markov Chain MC

Marcin Chrząszcz  
mchrzasz@cern.ch



University of  
Zurich <sup>UZH</sup>

Monte Carlo methods,  
21 April, 2016

There will be no lectures and class on 19<sup>th</sup> of May

## Trivial example

⇒ Lets start with a TRIVIAL example: we want to calculate  $S = A + B$ .  
We can rewrite it in:

$$S = p \frac{A}{p} + (1 - p) \frac{B}{1 - p}$$

and one can interpret the sum as expected value of:

$$W = \begin{cases} \frac{A}{p} & \text{with propability } p \\ \frac{A}{1-p} & \text{with propability } 1 - p \end{cases}$$

⇒ The algorithm:

- We generate a random variable  $W$  and calculate:

$$\hat{S} = \frac{1}{N} \sum_{i=1}^N W_i$$

⇒ The  $\hat{S}$  is an unbiased estimator of  $S$ .

## Trivial example2

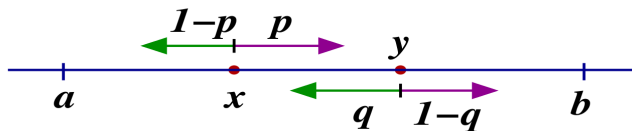
- Lets say we have a linear equation system:

$$\begin{aligned}X &= pY + (1 - p)A \\ Y &= qX + (1 - q)B\end{aligned}$$

- We know  $A, B, p, q$ ;  $X$  and  $Y$  are meant to be determined.
- Algorithm:
  1. We choose first element of the first equation with probability  $p$  and second with probability  $1 - p$ .
  2. We we choose the second one, the outcome of this MCMC is  $W = A$ .
  3. If we choose the first we go to second equation and choose the first element with probability  $q$  and the second with  $1 - q$ .
  4. We we choose the second one, the outcome of this MCMC is  $W = B$ .
  5. If we choose the first we go to the first equation back again.
  6. We repeat the procedure.
- We can estimate the solution of this system:

$$\hat{X} = \frac{1}{N} \sum_{i=1}^N W_i \quad \hat{\sigma}_X = \frac{1}{\sqrt{N-1}} \sqrt{\frac{1}{N} \sum_{i=1}^N W_i^2 - \hat{X}^2}$$

# Random walk



- ⇒ We are in the point  $x$  and we walk accordingly to the following rules:
- From point  $x$  we walk with probability  $p$  to point  $y$  or with  $1 - p$  to  $a$ .
  - From point  $y$  we walk with probability  $q$  to point  $x$  and with  $1 - Q$  to  $b$ .
  - The walks ends when you end up in  $a$  or  $b$ .
  - You get a "reward"  $A$  if you end up in point  $a$  and  $B$  if you end up in  $b$ .
  - $X$  is expected "reward" when you start the walk from  $x$ ,  $Y$  when you start from  $y$ .
- ⇒ The algorithm above is so-called random walk on the set  $\{a, x, y, b\}$
- ⇒ The described walked can solve the linear equation system that we discussed above.

# Markov Chain MC

- Consider a finite (or Countable set) possible states:  $S_1, S_2, \dots$
- The  $X_t$  is the state of the system in the time  $t$
- We are looking at discrete time steps: 1, 2, 3, ...
- At time  $t$  the state is denoted  $X_t$ .
- The conditional probability is defined as:

$$P(X_t = S_j | X_{t-1} = S_{j-1}, \dots, X_1 = S_1)$$

- The Markov chain is then if the probability depends only on previous step.

$$P(X_t = S_j | X_{t-1} = S_{j-1}, \dots, X_1 = S_1) = P(X_t = S_j | X_{t-1} = S_{j-1})$$

- For this reason this reason MCMC is also known as drunk sailor walk.
- Very powerful method. Used to solve linear eq. systems, invert matrix, solve differential equations, etc.
- Also used in physics problems: Brown motions, diffusion, etc.

# Linear equations system

⇒ Lets start from a linear equation system:

$$\mathbf{A}\vec{x} = \vec{b}, \quad \det \mathbf{A} \neq 0,$$

where  $\mathbf{A} = (a_{ij}, i, j = 1, 2, \dots, n)$  -matrix,  $\vec{b} = (b_1, b_2, \dots, b_n)$ -vector,  
 $\vec{x} = (x_1, x_2, \dots, x_n)$  - vector of unknowns.

⇒ The solution we mark as  $\vec{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$

⇒ The above system can be transformed into the iterative representation:

$$\vec{x} = \vec{a} + \mathbf{H}\vec{x}$$

where  $\mathbf{H}$  is a matrix,  $\vec{a}$  is a vector.

⇒ We assume that the matrix norm:

$$\|\mathbf{H}\| = \max_{1 \leq i \leq n} \sum_{j=1}^n |h_{ij}| < 1$$

# Linear equations system

⇒ Lets start from a linear equation system:

$$\mathbf{A}\vec{x} = \vec{b}, \quad \det \mathbf{A} \neq 0,$$

where  $\mathbf{A} = (a_{ij}, i, j = 1, 2, \dots, n)$  -matrix,  $\vec{b} = (b_1, b_2, \dots, b_n)$ -vector,  
 $\vec{x} = (x_1, x_2, \dots, x_n)$  - vector of unknowns.

⇒ The solution we mark as  $\vec{x}^0 = (x_1^0, x_2^0, \dots, x_n^0)$

⇒ The above system can be transformed into the iterative representation:

$$\vec{x} = \vec{a} + \mathbf{H}\vec{x}$$

where  $\mathbf{H}$  is a matrix,  $\vec{a}$  is a vector.

⇒ We assume that the matrix norm:

$$\|\mathbf{H}\| = \max_{1 \leq i \leq n} \sum_{j=1}^n |h_{ij}| < 1$$

⇒ We can always change transform every system to the iteration form:  $\mathbf{A} = \mathbf{V} - \mathbf{W}$ .

$$(\mathbf{V} - \mathbf{W})\vec{x} = \vec{b} \quad \mapsto \quad \vec{x} = \mathbf{V}^{-1}\vec{b} + \mathbf{V}^{-1}\mathbf{W}\vec{x}$$



# Linear equations system

⇒ Now we further modify the equation system:

$$\vec{x} = \vec{a} + \mathbf{H}\vec{x} \Rightarrow (\mathbf{I} - \mathbf{H})\vec{x} = \vec{a}$$

where  $\mathbf{I} = \delta_{ij}$  - unit matrix,  $\delta_{ij}$  is the Kronecker delta.

⇒ What one can do is to represent the solution in terms of Neumann series:

$$\vec{x}^0 = (\mathbf{I} - \mathbf{H})^{-1}\vec{a} = \vec{a} + \mathbf{H}\vec{a} + \mathbf{H}^2\vec{a} + \mathbf{H}^3\vec{a} + \dots$$

⇒ So for the  $i^{\text{th}}$  component we have:

$$\begin{aligned} x_i^0 &= a_i + \sum_{j=1}^n h_{ij} a_j + \sum_{j_1=1}^n \sum_{j_2=1}^n h_{ij_1} h_{j_1 j_2} a_{j_2} \\ &+ \dots + \sum_{j_1=1}^n \dots \sum_{j_n=1}^n h_{ij_1} \dots h_{j_{n-1} j_n} a_{j_n} \end{aligned}$$

⇒ We will construct a probabilistic interpretation using MCMC and then we show that the expected value is equal to the above formula.

# Neumann-Ulam method

- To do so we add to our matrix an additional column of the matrix:

$$h_{i,0} = 1 - \sum_{j=1}^n h_{ij} > 0$$

- The system has states:  $\{0, 1, 2, \dots, n\}$
- State at  $t$  time is denoted as  $i_t$  ( $i_t = 0, 1, 2, \dots, n; t = 0, 1, \dots$ )
- We make a random walk accordingly to the following rules:
  - At the beginning of the walk ( $t = 0$ ) we are at  $i_0$ .
  - In the  $t$  moment we are in the  $i_t$  position then in  $t + 1$  time stamp we move to state  $i_{t+1}$  with the probability  $h_{i_t i_{t+1}}$ .
  - We stop walking if we are in state 0.
- The path  $\gamma = (i_0, i_1, i_2, \dots, i_k, 0)$  is called trajectory.
- For each trajectory we assign a number:

$$X(\gamma) = X(i_0, i_1, i_2, \dots, i_k, 0) = \frac{a_{i_k}}{h_{i_k 0}}$$

# Neumann-Ulam method

- ⇒ The  $X(\gamma)$  variable is a random variable from:  $\{a_1/h_{1,0}, a_2/h_{2,0}, \dots, a_n/h_{n,0}\}$ .  
The probability that  $X(\gamma) = a_j/h_{j,0}$  is equal to the probability that the last non zero state of the  $\gamma$  trajectory is  $j$ .
- ⇒ The expected value of the  $X(\gamma)$  trajectory if the trajectory begins from  $i_0 = s$  is:

$$E\{X(\gamma)|i_0 = s\} = \sum_{k=0}^{\infty} \sum_{\{\gamma_k\}} X(\gamma)P(\gamma)$$

where  $\gamma_k$  is a trajectory of length  $k$ , which starts in  $i_0 = s$  and  $P(\gamma)$  is the probability of occurrence of this trajectory. ⇒ Yes you guess it lets do Taylor expansion:

$$E\{X(\gamma)|i_0 = s\} = \sum_{\gamma_0} X(\gamma)P(\gamma) + \sum_{\gamma_1} X(\gamma)P(\gamma) + \dots + \sum_{\gamma_k} X(\gamma)P(\gamma)$$

- ⇒ Now let's examine the elements of the above series.

# Neumann-Ulam method

$\{\gamma_0\}$ : One trajectory:  $\gamma_0 = (i_0 = s|0)$ ,  $P(\gamma_0) = h_{s,0}$  and  $X(\gamma_0) = a_s/h_{s,0}$ . So:

$$\sum_{\gamma_0} X(\gamma)P(\gamma) = \frac{a_s}{h_{s,0}} h_{s,0} = a_s$$

$\{\gamma_1\}$ : Trajectories:  $\gamma_1 = (i_0 = s, i_1|0)$ ,  $i_1 \neq 0$ ,  $P(\gamma_1) = P(s, i_1, 0) = h_{s,i_1} h_{i_1,0}$  and  $X(\gamma_1) = a_{i_1}/h_{i_1,0}$ . So:

$$\sum_{\gamma_1} X(\gamma)P(\gamma) = \sum_{i_1=1}^n \frac{a_{i_1}}{h_{i_1,0}} h_{s,i_1} h_{i_1,0} = \sum_{i_1=1}^n h_{s,i_1} a_{i_1}$$

$\{\gamma_2\}$ : Trajectories:  $\gamma_2 = (i_0 = s, i_1, i_2|0)$ ,  $i_1, i_2 \neq 0$ ,  
 $P(\gamma_2) = P(s, i_1, i_2, 0) = h_{s,i_1} h_{i_1,i_2} h_{i_2,0}$  and  $X(\gamma_2) = a_{i_2}/h_{i_2,0}$ . So:

$$\sum_{\gamma_2} X(\gamma)P(\gamma) = \sum_{i_1=1}^n \sum_{i_2=1}^n \frac{a_{i_2}}{h_{i_2,0}} h_{s,i_1} h_{i_1,i_2} h_{i_2,0} = \sum_{i_1=1}^n \sum_{i_2=1}^n h_{s,i_1} h_{i_1,i_2} a_{i_2}$$

⇒ etc...

# Neumann-Ulam method

⇒ After summing up:

$$\begin{aligned} E\{X(\gamma)|i_0 = s\} &= a_s + \sum_{i_1=1}^n h_{s,i_1} a_{i_1} + \sum_{i_1=1}^n \sum_{i_2=1}^n h_{s,i_1} h_{i_1,i_2} a_{i_2} + \dots \\ &+ \sum_{i_1=1}^n \sum_{i_2=1}^n \dots \sum_{i_k=1}^n h_{s,i_1} h_{i_1,i_2} \dots h_{i_{k-1},i_k} a_{i_k} + \dots \end{aligned}$$

⇒ If you compare this expression with the Neumann series we will they are the same so:

$$x_i^0 = E\{X(\gamma)|i_0 = i\}$$

To sum up:

We have proven that solving a linear system can be represented by an expectation value of the random variable  $X(\gamma)$ . The error is computed using standard deviation equation.

# Neumann-Ulam method

- For example lets try to solve this equation system:

$$\vec{x} = \begin{pmatrix} 1.5 \\ -1.0 \\ 0.7 \end{pmatrix} + \begin{pmatrix} 0.2 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.2 \\ 0.3 & 0.1 & 0.1 \end{pmatrix} \vec{x}$$

- The solution is  $\vec{x}_0 = (2.154303, 0.237389, 1.522255)$ .
- The propability matrix  $h_{ij}$  has the shape:

$i/j$	1	2	3	0
1	0.2	0.3	0.1	0.4
2	0.4	0.3	0.2	0.1
3	0.3	0.1	0.1	0.5

- An example solution:

```
mchraszcz-ThinkPad-W530% ./mark.x 1 1000000  
2.15625
```

# Neumann-Ulam dual method

- The problem with Neumann-Ulam method is that you need to repeat it for each of the coordinates of the  $\vec{x}_0$  vector.
- The dual method calculates the whole  $\vec{x}_0$  vector.
- The algorithm:
  - On the indexes:  $\{0, 1, \dots, n\}$  we set a probability distribution:  $q_1, q_2, \dots, q_n, q_i > 0$  and  $\sum_{i=1}^n q_i = 1$ .
  - The starting point we select from  $q_i$  distribution.
  - If in  $t$  time we are in  $i_t$  state then with probability  $p(i_{t+1}|i_t) = h_{i_{t+1}, i_t}$  in  $t + 1$  we will be in state  $i_{t+1}$ . For  $i_{t+1} = 0$  we define the probability:  $h_{0, i_t} = 1 - \sum_{j=1}^n h_{j, i_t}$ . Here we also assume that  $h_{j, i_t} > 0$ .
  - NOTE: there the matrix is transposed compared to previous method:  $H^T$ .
  - Again we end our walk when we are at state 0.
  - For the trajectory:  $\gamma = (i_0, i_1, \dots, i_k, 0)$ , we assign the vector:

$$\vec{Y}(\gamma) = \frac{a_{i_0}}{q_{i_0} p(0|i_k)} \hat{e}_{i_k} \in \mathcal{R}^n$$

- The solution will be :  $\vec{x}^0 = \frac{1}{N} \sum \vec{Y}(\gamma)$

# Neumann-Ulam dual method, proof

⇒ If  $Y_i(\gamma)$  is the  $i$ -th component of the  $\vec{Y}(\gamma)$  vector. One needs to show:

$$E\{Y_i(\gamma)\} = x_j^0$$

⇒ From definition:

$$Y_j(i_1, \dots, i_k, 0) = \begin{cases} \frac{a_{i_k}}{q_{i_0} p(0|i_k)} & i_k = j \\ 0 & i_k \neq j \end{cases}$$

⇒ The expected value:

$$E\{Y_j(\gamma)\} = \sum_{\text{trajectories}} \frac{a_j}{q_{i_0} p(0|i_k)} P(i_1, i_2, \dots, i_k, 0),$$

where  $P(i_1, i_2, \dots, i_k, 0)$  is the probability of this trajectory occurring.

⇒ But by our definition the probability:

$$P(i_0, i_1, \dots, i_{k-1}, j, 0) = q_0 h_{i_1, i_0} \dots h_{k, i_{k-1}} p(0|j)$$

⇒ In the end we get:

$$E(Y_j(\gamma)) = \sum_{k=0}^{\infty} \sum_{i_{k-1}=1}^n \dots \sum_{i_1=1}^n \sum_{i_0=1}^n h_{j, i_{k-1}} h_{j, i_{k-1}} \dots h_{i_2, i_1} h_{i_1, i_0} a_{i_0}$$



# Neumann-Ulam dual method

- Let's try to solve the equation system:

$$\vec{x} = \begin{pmatrix} 1.5 \\ -1.0 \\ 0.7 \end{pmatrix} + \begin{pmatrix} 0.2 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.2 \\ 0.1 & 0.1 & 0.1 \end{pmatrix} \vec{x}$$

- The solution is:  $\vec{x}_0 = (2.0, 0.0, 1.0)$ .
- Let's put the initial probability as constant:

$$q_1 = q_2 = q_3 = \frac{1}{3}$$

- The propability matrix  $h_{ij}$  has the shape:

$i/j$	1	2	3	4
1	0.2	0.4	0.1	0.3
2	0.3	0.3	0.1	0.3
3	0.1	0.2	0.1	0.6

- An example solution:

```
mchrzasz-ThinkPad-W530% ./mark2.x 1000000  
1.9943 0.001806 1.00267
```

## Generalization

⇒ Up to now we assumed that each of the matrix elements  $h_{i,j} \geq 0$ .  
Now if this is not true:

⇒ We take a probability matrix  $P = p_{ij}$  such that:

$$p_{ij} \geq 0 \quad p_{ij} = 0 \Leftrightarrow h_{ij} = 0, \quad p_{i,0} = 1 - \sum_j p(i, j) > 0.$$

⇒ To solve the system we construct a Markov Chain with the  $P$  matrix as probabilities of transitions.

⇒ The probability of a trajectory is equal ( $i_0 = i$ ):

$$P(\gamma_k) = p_{i,i_1} p_{i_1,i_2} \cdots p_{i_k,0}$$

⇒ The trajectory we assign the number:

$$X(\gamma_k) = \nu_{i,i_1} \nu_{i_1,i_2} \cdots \nu_{i_{k-1},i_k} \frac{a_{i_k}}{p_{i_k,0}}$$

where

$$\nu_{i,j} = \begin{cases} h_{ij}/p_{ij}, & p_{ij} \neq 0 \\ 1 & p_{ij} = 0 \end{cases}$$

## Generalization, proof

⇒ For a  $X(\gamma)$  trajectory the expected value is:

$$E\{X(\gamma_k)\} = \sum_{k=0}^{\infty} \sum_{\gamma_k} X(\gamma_k) P\{X(\gamma_k)\}$$

⇒ The probability is given by the formula:

$$\begin{aligned} P\{X(\gamma_k)\} &= P\{X(\gamma_k) = \nu_{i,i_1} \nu_{i_1,i_2} \dots \nu_{i_{k-1},i_k} \frac{a_{i_k}}{p_{i_k,0}}\} \\ &= p_{i,i_1} \dots p_{i_{k-1},i_k} p_{i_k,0} \end{aligned}$$

⇒ However:

$$X(\gamma_k) P\{X(\gamma_k)\} = h_{i,i_1} h_{i_1,i_2} \dots h_{i_{k-1},i_k} a_{i_k}$$

so:

$$\sum_{\gamma_k} X(\gamma_k) P\{X(\gamma_k)\} = \sum_{i_1=1} \dots \sum_{i_k=1} h_{i,i_1} h_{i_1,i_2} \dots h_{i_{k-1},i_k} a_{i_k}$$

# Generalization, the algorithm

- ⇒ We set the  $P$  matrix in a arbitrary way.
- ⇒ If in the  $t$  moment the point is in the  $i_t$  state, then with the probability  $p_{i_t, i_{t+1}}$  he can go to  $i_{t+1}$  state.
- ⇒ We stop the walk once we reach 0.
- ⇒ For the given trajectory we assign the value:  $X(\gamma_k)$
- ⇒ We repeat the procedure  $N$  times and take the mean and RMS.
- ⇒ We repeat this also for every of the  $\vec{x}^0$  vector components.

# Wasow method

⇒ The main problem with the Neumann-Ulam methods is the fact that each time we estimate only one of the part of the taylor expansion.

⇒ W.Wasow (1956) was smarter:

- For the trajectory:  $\gamma(i_0, i_1, \dots, i_k, 0)$  we look trajectories:

$$(i_0), (i_0, i_1), (i_0, i_1, \dots, i_k)$$

and for each we associate a number:

$$(i_0, i_1, i_2, \dots, i_m), 0 \leq m \leq k$$

we assign a number:

$$\nu_{i_0, i_1} \nu_{i_1, i_2} \dots \nu_{i_{m-1}, i_m} a_{i_m}$$

⇒ For the trajectory we define:

$$X^*(\gamma) = \sum_{m=0}^k \nu_{i_1, i_2} \dots \nu_{i_{m-1}, i_m} a_{i_m}$$

⇒ One can proof that:

$$E\{X^*(\gamma) | i_0 = i\} = x_i^0$$

# Homework

⇒ E8.(1,2,3) Please solve the linear equation system from slide 16 using Wasow, Neumann-Ulam, dual Neumann-Ulam methods.

# Backup