Applications of Monte Carlo methods

Marcin Chrząszcz
mchrzasz@cern.ch

University of Zurich[UZH]

Experimental Methods in Particle Physics,
26 November, 2015

Markov Chain MC

- Consider a finite possible states: S_1, S_2, ...
- And the time steps of time, labelled as 1, 2, ...
- At time t the state is denoted X_t.
- The conditional probability is defined as:

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

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

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

- For this reason this reason MCMC is also knows as drunk sailor walk.
- Very powerful method. Used to solve linear eq. systems, invert matrix, solve differential equations, etc.

Linear Equations

- 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:

$$\hat{X} = \frac{1}{N} \sum_{i=1} W_i \qquad \hat{\sigma_X} = \frac{1}{\sqrt{N-1}} \sqrt{\frac{1}{N} \sum_{i=1}^{N} W_i^2 - \hat{X}^2}$$

Neumann-Ulam method

- Let's try apply the basic MCMC method to solve a simple linear equation system:

$$A\overrightarrow{x} = \overrightarrow{b}$$

- The above system can be (always, see linear algebra lecture) translated into system:

$$\overrightarrow{x} = \overrightarrow{a} + H\overrightarrow{x}$$

- For this method we assume that the norm of the matrix is:

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

- Which we can write in a form:

$$(1 - H)\overrightarrow{x} = \overrightarrow{a}$$

Neumann-Ulam method

- The solution would be then:

$$\overrightarrow{x}_0 = (1 - H)^{-1} \overrightarrow{a}$$

- We can Taylor expend this:

$$\overrightarrow{x}_0 = (1 - H)^{-1} \overrightarrow{a} = \overrightarrow{a} + H \overrightarrow{a} + H^2 \overrightarrow{a} + H^3 \overrightarrow{a} + ...$$

- For the i-th component of the $\overrightarrow{x}$ vector:

$$x_0^i = a_i + \sum_{j=1}^{n} h_{ij} a_{j_1} + \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} h_{ij_1} h_{ij_2} a_{j_2} + \sum_{j_1=1}^{n} \sum_{j_2=1}^{n} \sum_{j_3=1}^{n} h_{ij_1} h_{ij_2} h_{ij_3} a_{j_3} + ...$$

- One can construct probabilistic behaviour of a system that follows the path of equation above.

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: $\{0, 1, 2..., n\}$
- State at t time is denoted as i_t.
- We make a random walk accordingly to to the following rules:
  - At the begging 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 $X(\gamma) = (i_0, i_1, i_2, ..., i_k, 0)$ is called trajectory.
- It can be proven that $x_i^0 = E\{X(\gamma)|i_0 = j\}$.

- 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	4
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:

```
mchrzasz-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 $\overrightarrow{x}_0$ vector.
- The dual method calculates the whole $\overrightarrow{x}_0$ vector.
- The algorithm:
  - On the indexes: $\{0, 1, ..., n\}$ we set a probability distribution: $q_1, q_2, ..., q_n$, $q_i > 0$ 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_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: $\gamma = (i_0, i_1, ..., i_k, 0)$, we assign the vector:
$$\overrightarrow{Y}(\gamma) = \frac{a_{i_0}}{q_{i_0} p(0|i_k)} \widehat{e}_{i_k} \in \mathcal{R}^n$$
- The solution will be : $\overrightarrow{x}^0 = \frac{1}{N} \sum \overrightarrow{Y}(\gamma)$

- 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
```

Look elsewhere effect, Lecture3/LEE

- Look elsewhere effect addresses the following problem:
    - Imagine you observed a 3σ deviation in one of the observable that you measured.
    - Before you get excited one needs to understand if given the fact that you had so many measurements this might happen!
- Example: Let's say we have measured 50 observables. What is the probability to observed 1 that is 3σ away from theory prediction?
- Let's simulate 50 Gaussian distribution centred at 0 and width of 1. We count how simulations where at least one of the 50 numbers have the absolute value > 3.
- More complicated example: what if you observed 3 in a row 2σ fluctuations among 50 measurements?
- This kind of studies are the best solvable by MC simulations.

Travelling Salesman Problem

- Salesman starting from his base has to visit $n-1$ other locations and return to base headquarters. The problem is to find the shortest way.
- For large n the problem can't be solver by brutal force as the complexity of the problem is $(n-1)!$
- There exist simplified numerical solutions assuming factorizations. Unfortunately even those require anonymous computing power.
- Can MC help? YES :)
- The minimum distance l has to depend on 2 factors: P the area of the city the Salesman is travelling and the density of places he wants to visit: $\frac{n}{P}$
- Form this we can assume:

$$l \sim P^a (\frac{n}{P})^b = P^{a-b} n^b.$$

Traveling Salesman Problem

- From dimension analysis:

$$a - b = \frac{1}{2}.$$

- To get l we need square root of area.
- From this it's obvious:

$$l \sim P^a (\frac{n}{P})^b = P^{0.5} n^{a-0.5}.$$

- Now we can multiply the area by alpha factor that keeps the density constant then:

$$l \sim \alpha^0 .5\alpha 6a - 0.5 = \alpha^a$$

- In this case the distance between the clients will not change, but the number of clients will increase by α so:

$$l \sim \alpha$$

- In the end we get: $a = 1$

Traveling Salesman Problem

- In total:

$$l \sim k(nP)^{0.5}$$

- Of course the k depends on the shape of the area and locations of client. However for large n the k starts loosing the dependency. It's an asymptotically free estimator.
- To use the above formula we need to somehow calculate k.
- How to estimate this? Well make a TOY MC: take a square put uniformly n points. Then we can calculate l. Then it's trivial:

$$k = l(nP)^{-0.5}$$

Traveling Salesman Problem

- This kind of MC experiment might require large CPU power and time. The adventage is that once we solve the problem we can use the obtained k for other cases (it's universal constant!).
- It turns out that:

$$k \sim \frac{3}{4}$$

- Ok, but in this case we can calculate l but not the actual shortest way! Why the hell we did this exercise?!
- Turns out that for most of the problems we are looking for the solution that is close to smallest l not the exact minimum.

War Games

- S. Andersoon 1966 simulated for Swedish government how would a tank battle look like.
- Each of the sides has 15 tanks. that they allocate on the battle field.
- The battle is done in time steps.
- Each tank has 5 states:
  - OK
  - Tank can only shoot
  - Tank can only move
  - Tank is destroyed
  - Temporary states
- This models made possible to check different fighting strategies.

Q & A

Backup