Matrix inversion and Partial Differential Equation Solving

Marcin Chrząszcz
mchrzasz@cern.ch

University of Zurich[UZH]

Monte Carlo methods,
28 April, 2016

Announcement

There will be no lectures and class on 19^{th} of May

Matrix inversion

⇛ The last time we discussed the method of linear equations solving. The same methods can be used for matrix inversions! The columns of inverse matrix can be found solving:

$$\mathbf{A}\overrightarrow{x} = \hat{e}_i, \quad i = 1, 2, ..., n$$

⇛ In order to determine the inverse of a matrix **A** we need to choose a temprorary matrix **M** such that:

$$\mathbf{H} = \mathbf{I} - \mathbf{MA}$$

with the normalization condition:

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

where **I** is a unit matrix.

⇛ Next we Neumann expand the $(\mathbf{MA})^{-1}$ matrix:

$$(\mathbf{MA})^{-1} = (\mathbf{I} - \mathbf{H})^{-1} = \mathbf{I} + \mathbf{H} + \mathbf{H}^2 + ...$$

⇛ The inverse matrix we get from the equation:

$$\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{M}^{-1}\mathbf{M} = (\mathbf{MA})^{-1}\mathbf{M}$$

Matrix inversion, basic method

$\Rrightarrow$ For the (i, j) element of the matrix $(MA)^{-1}$ we have:

$$(MA)^{-1}_{ij} = \delta_{ij} + h_{ij} + \sum_{i_1=1}^{n} h_{ii_1} h_{i_1 j} + \sum_{i_1=1}^{n} \sum_{i_2=1}^{n} h_{ii_1} h_{i_1 i_2} h_{i_2 j} + ...$$

$\Rrightarrow$ The algorithm: We choose freely a probability matrix $P = (p_{ij})$ with the conditions:

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

$\Rrightarrow$ We construct a random walk for the state set $\{0, 1, 2, 3..., n\}$:

1. In the initial moment $(t = 0)$ we start in the state $i_0 = i$.
2. If in the moment t the point is in the i_t state, then in the time $t + 1$ he will be in state i_{t+1} with the probability $p_{i_t, t_{t+1}}$.
3. We stop the walk if we end up in the state 0.

Matrix inversion, basic method

⇛ For the observed trajectory $\gamma_k = (i, i_1, .., j_k, 0)$ we assign the value of:

$$X(\gamma_k) = \frac{h_{ii_1} h_{i_1 i_2} ... h_{i_{k-1} i_k}}{p_{ii_1} p_{i_1 i_2} ... p_{i_{k-1} i_k}} \frac{\delta_{i_k j}}{p_{i_k 0}}$$

⇛ The mean is the of all observed $X(\gamma_k)$ is an unbiased estimator of the $(MA)^{-1}_{ij}$.

Prove:

- The probability of observing the γ_k trajectory:

$$P(\gamma_k) = p_{ii_1} p_{i_1 i_2} ... p_{i_{k-1} i_k} p_{i_k 0}$$

- Form this point we follow the prove of the previous lecture (Neumann-Ulan) and prove that:

$$E\{X(\gamma_k)\} = (MA)^{-1}$$

⇛ A different estimator for the $(MA)^{-1}_{ij}$ element is the Wasow estimator:

$$X^*(\gamma_k) = \sum_{m=0}^{k} \frac{h_{ii_1} h_{i_1 i_2} ... h_{i_{m-1} i_m}}{p_{ii_1} p_{i_1 i_2} ... p_{i_{m-1} i_m}} \delta_{i_m j}$$

Matrix inversion, dual method

⇛ On the set of states $\{0, 1, 2, ..., n\}$ we set a binned p.d.f.

$$q_1, q_2, ..., q_n \text{ such that } q_i > 0,\ i = 1, 2, 3...n \text{ and } \sum_{i=1}^{n} q_i = 1.$$

⇛ Then choose arbitrary the probability matrix P (usual restrictions apply):

- The initial point we choose with the probability q_i.
- If in the moment t the point is in the i_t state, then in the time $t+1$ he will be in state i_{t+1} with the probability $p_{i_t, t_{t+1}}$.
- The walk ends when we reach 0 state.
- For the trajectory we assign a matrix:

$$Y(\gamma_k) = \frac{h_{i_1 i} h_{i_2 i_1} ... h_{i_k i_{k-1}}}{p_{i_1 i} p_{i_2 i_1} ... p_{i_k i_{k-1}}} \frac{1}{q_{i_0} p_{i_k 0}} e_{i_k i_0} \in \mathbb{R}^n \times \mathbb{R}^n$$

⇛ The mean of $Y(\gamma)$ is an unbiased estimator of the $(MA)^{-1}$ matrix.

⇛ The Wasow estimator reads:

$$Y^* = \sum_{m=0}^{k} \frac{h_{i_1 i} h_{i_2 i_1} ... h_{i_m i_{m-1}}}{p_{i_1 i} p_{i_2 i_1} ... p_{i_m i_{m-1}}} e_{i_m i_0} \in \mathbb{R}^n \times \mathbb{R}^n$$

Partial differential equations, intro

⇛ Let's say we are want to describe a point that walks on the $\mathbb{R}$ axis:

- At the beginning $(t = 0)$ the particle is at $x = 0$
- If in the t the particle is in the x then in the time $t + 1$ it walks to $x + 1$ with the known probability p and to the point $x - 1$ with the probability $q = 1 - p$.
- The moves are independent.

⇛ So let's try to described the motion of the particle.
⇛ The solution is clearly a probabilistic problem. Let $\nu(x, t)$ be a probability that at time t particle is in position x. We get the following equation:

$$\nu(x, t + 1) = p\nu(x - 1, t) + q\nu(x + 1, t)$$

with the initial conditions:

$$\nu(0, 0) = 1, \qquad \nu(x, 0) = 0 \;\text{ if } x \neq 0.$$

⇛ The above functions describes the whole system (every (t, x) point).

Partial differential equations, intro

⇛ Now in differential equation language we would say that the particle walks in steps of Δx in times: $k\Delta t$, $k = 1, 2, 3 \ldots$.:

$$\nu(x, t + \Delta t) = p\nu(x - \Delta x, t) + q\nu(x + \Delta x, t).$$

⇛ To solve this equation we need to expand the $\nu(x, t)$ funciton in the Taylor series:

$$\nu(x,t) + \frac{\partial \nu(x,t)}{\partial t}\Delta t = p\nu(x,t) - p\frac{\partial \nu(x,t)}{\partial x}\Delta x + \frac{1}{2}p\frac{\partial^2 \nu(x,t)}{\partial x^2}(\Delta x)^2$$
$$+q\nu(x,t) + q\frac{\partial \nu(x,t)}{\partial x}\Delta x + \frac{1}{2}q\frac{\partial^2 \nu(x,t)}{\partial x^2}(\Delta x)^2$$

⇛ From which we get:

$$\frac{\partial \nu(x,t)}{\partial t}\Delta t = -(p-q)\frac{\partial \nu(x,t)}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 \nu(x,t)}{\partial x^2}(\Delta x)^2$$

⇛ Now We divide the equation by Δt and take the $\Delta t \rightarrow 0$:

$$(p-q)\frac{\Delta x}{\Delta t} \rightarrow 2c, \qquad \frac{(\Delta x)^2}{\Delta t} \rightarrow 2D,$$

⇛ We get the Fokker-Planck equation for the diffusion with current:

$$\frac{\partial \nu(x,t)}{\partial t} = -2c\frac{\partial \nu(x,t)}{\partial x} + D\frac{\partial^2 \nu(x,t)}{\partial x^2}$$

⇛ The D is the diffusion coefficient, c is the speed of current. For $c = 0$ it is a symmetric distribution.

Laplace equation, Dirichlet boundary conditions

⇛ The aforementioned example shows the way to solve the partial differential equation using Markov Chain MC.
⇛ We will see how different classes of partial differential equations can be approximated with a Markov Chain MC, whose expectation value is the solution of the equation.
⇛ The Laplace equation:

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + ... + \frac{\partial^2 u}{\partial x_k^2} = 0$$

The $u(x_1, x_2, ..., x_k)$ function that is a solution of above equation we call harmonic function. If one knows the values of the harmonic function on the edges $\Gamma(D)$ of the D domain one can solve the equation.

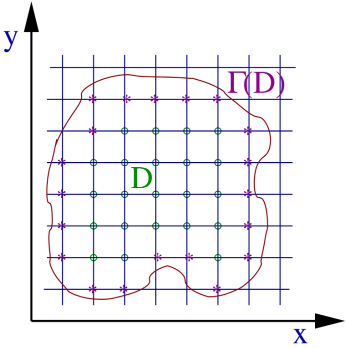
The Dirichlet boundary conditions:

Find the values of $u(x_1, x_2, ..., x_k)$ inside the D domain knowing the values of the edge are given with a function:

$$u(x_1, x_2, ..., x_k) = f(x_1, x_2, ..., x_k) \in \Gamma(D)$$

⇛ Now I am lazy so I put $k = 2$ but it's the same for all k!

Laplace equation, Dirichlet boundary conditions



⇛ We will put the Dirichlet boundary condition as a discrete condition:

- The domain D we put a lattice with distance h.
- Some points we treat as inside (denoted with circles). Their form a set denoted D^*.
- The other points we consider as the boundary points and they form a set $\Gamma(D)$.

⇛ We express the second derivatives with the discrete form:

$$\frac{\frac{u(x+h)-u(x)}{h}-\frac{u(x)-u(x-h)}{h}}{h}=\frac{u(x+h)-2u(x)+u(x-h)}{h^2}$$

⇛ Now we choose the units so $h=1$.

Laplace equation, Dirichlet boundary conditions

The Dirichlet condition in the discrete form:

Find the u^* function which obeys the differential equation:

$$U^*(x,y) = \frac{1}{4}\left[u^*(x-1,y) + u^*(x+1,y) + u^*(x,y-1) + u^*(x,y+1)\right]$$

in all points $(x,y) \in D^*$ with the condition:

$$u^*(x,y) = f^*(x,y), \quad (x,y) \in \Gamma(D^*)$$

where $f^*(x,y)$ is the discrete equivalent of $f(x,y)$ function.

$\Rightarrow$ We consider a random walk over the lattice $D^* \cup \Gamma(D^*)$.

- In the $t = 0$ we are in some point $(\xi, \eta) \in D^*)$
- If at the t the particle is in (x,y) then at $t+1$ it can go with equal probability to any of the four neighbour lattices: $(x-1,y)$, $(x+1,y)$, $(x,y-1)$, $(x,y+1)$.
- If the particle at some moment gets to the edge $\Gamma(D^*$ then the walk is terminated.
- For the particle trajectory we assign the value of: $\nu(\xi,\eta) = f^*(x,y)$, where $(x,y) \in \Gamma(D^*)$.

Laplace equation, Dirichlet boundary conditions

⇛ Let $p_{\xi,\eta}(x,y)$ be the probability of particle walk that starting in (ξ,η) to end the walk in (x,y).

⇛ The possibilities:

1. The point $(\xi,\eta) \in \Gamma(D^*)$. Then:

$$p_{\xi,\eta}(x,y) = \begin{cases} 1, & (x,y) = \xi,\eta) \\ 0, & (x,y) \neq \xi,\eta) \end{cases} \tag{1}$$

2. The point $(\xi,\eta) \in D^*$:

$$p_{\xi,\eta}(x,y) = \frac{1}{4}\left[p_{\xi-1,\eta}(x,y) + p_{\xi+1,\eta}(x,y) + p_{\xi,\eta-1}(x,y) + p_{\xi,\eta+1}(x,y)\right] \tag{2}$$

this is because to get to (x,y) the particle has to walk through one of the neighbours: $(x-1,y)$, $(x+1,y)$, $(x,y-1)$, $(x,y+1)$.

⇛ The expected value of the $\nu(\xi,\eta)$ is given by equation:

$$E(\xi,\eta) = \sum_{(x,y)\in\Gamma^*} p_{\xi,\eta}(x,y) f^*(x,y) \tag{3}$$

where the summing is over all boundary points

Laplace equation, Dirichlet boundary conditions

⇛ Now multiplying the 2 by $f^*(x,y)$ and summing over all edge points (x,y):

$$E(\xi,\eta) = \frac{1}{4}\left[E(\xi-1,\eta) + E(\xi+1,\eta) + E(\xi,\eta-1) + E(\xi,\eta+1)\right]$$

⇛ Putting now 1 to 3 one gets:

$$E(x,y) = f^*(x,y), \quad (\xi,\eta) \in \Gamma(D^*)$$

⇛ Now the expected value solves identical equation as our $u^*(x,y)$ function. From this we conclude:

$$E(x,y) = u^*(x,y)$$

⇛ The algorithm:

- We put a particle in (x,y).
- We observe it's walk up to the moment when it's on the edge $\Gamma(D^*)$.
- We calculate the value of f^* function in the point where the particle stops.
- Repeat the walk N times taking the average afterwards.

Important:

One can show the the error does not depend on the dimensions!

Example

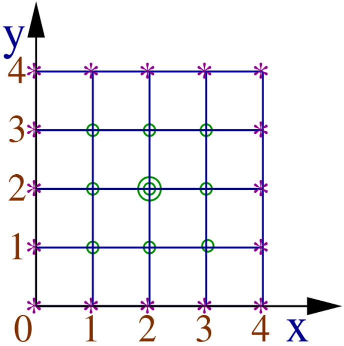
Let function $u(x,y)$ be a solution of Laplace equation in the square: $0 \leqslant x, y \leqslant 4$ with the boundary conditions:

$$u(x,0) = 0, \quad u(4,y) = y, \quad u(x,4) = x, \quad x(0,y) = 0$$

$\Rightarrow$ Find the $u(2,2)$!

$\Rightarrow$ The exact solution: $u(x,y) = xy/4$ so $u(2,2) = 1$.

- We transform the continues problem to a discrete one with $h = 1$.
- Perform a random walk starting from $(2,2)$ which ends on the edge assigning as a result the appropriative values of the edge conditions as an outcome.



$\Rightarrow$ E9.1 Implement the above example and find $u(2,2)$.

Parabolic equation

⇛ We are looking for a function $u(x_1, x_2, ..., x_k, t)$, which inside the $D \subset \mathbb{R}^k$ obeys the parabolic equation:

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + ... + \frac{\partial^2 u}{\partial x_k^2} = c\frac{\partial u}{\partial t}$$

with the boundary conditions:

$$u(x_1, x_2, ..., x_k, t) = g(x_1, x_2, ..., x_k, t), \quad (x_1, x_2, x_3, ..., x_k) \in \Gamma(D)$$

and with the initial conditions:

$$u(x_1, x_2, ..., x_k, 0) = h(x_1, x_2, ..., x_k, t), \quad (x_1, x_2, x_3, ..., x_k) \in D$$

⇛ In the general case the boundary conditions might have also the derivatives.

⇛ We will find the solution to the above problem using random walk starting from 1-dim case and then generalize it for n-dim.

Parabolic equation, 1-dim

⇛ We are looking for a function $u(x,t)$, which satisfies the equation:

$$\frac{\partial^2 u}{\partial x^2} = c\frac{\partial u}{\partial t}$$

with the boundary conditions:

$$u(0,t) = f_1(t), \;\; u(a,t) = f_2(t)$$

and with the initial conditions:

$$u(x,0) = g(x).$$

⇛ The above equation can be seen as describing the temperature of a line with time. We know the initial temperature in different points and we know that the temperature on the end points is know.

⇛ The above problem can be discreteized:

$$x = kh, \;\; h = \frac{a}{n}, \;\; k = 1, 2, ...n \qquad t = jl, \;\; j = 0, 1, 2, 3..., \;\; l = \mathrm{const}$$

⇛ The differential equation:

$$\frac{u(x+h,t-l) - 2u(x,t-l) + u(x-h,t-l}{h^2}) = c\frac{u(x,t) - u(x,t-l)}{l}$$

Parabolic equation, 1-dim

⇛ The steps we choose such that: $ch^2 = 2l$.
⇛ Then we obtain the equation:

$$u(x,t) = \frac{1}{2}u(x+h,t-l) + \frac{1}{2}u(x-h,t-l)$$

⇛ The value of function u in the point x and t can be evaluated with the arithmetic mean form points: $x+h$ and $x-h$ in the previous time step. ⇛ The algorithm estimating the function in the time τ and point ξ:

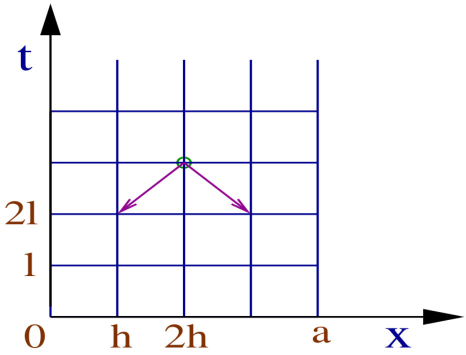
- The particle we put in the point ξ and a "weight" equal τ.
- If in a given time step t particle is at x then with $50:50$ chances it can go to $x-h$ or $x+h$ and time $t-l$.
- The particle ends the walk in two situations:
  - If it reaches the $x=0$ or $x=a$. In this case we assign to a given trajectory a value of $f_1(t)$ or $f_2(t)$, where t is the actuall "weight".
  - If the "weight" of the particle is equal zero. in this case we assign as a value of the trajectory the $g(x)$, where x is the actual position of the particle.

Parabolic equation, 1-dim

⇛ Repeat the above procedure N times. The expected value of a function u in (ξ, τ) point is the mean of observed values.

Digression:

The 1-dim calse can be treated as a 2-dim (x, t), where the area is unbounded in the t dimension. The walk is terminated after maximum τ/l steps.

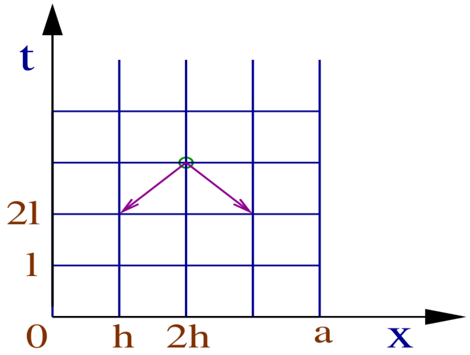


Parabolic equation, 1-dim

⇛ Repeat the above procedure N times. The expected value of a function u in (ξ, τ) point is the mean of observed values.

Digresion:

The 1-dim calse can be treated as a 2-dim (x, t), where the area is unbounded in the t dimension. The walk is terminated after maximum τ / l steps.



Parabolic equation, n-dim generalization

⇛ We still choose the k and l values accordingly to:

$$\frac{ch^2}{l} = 2k$$

where k is the number of space dimensions.

⇛ We get:

$$u(x_1, x_2, ..., x_k) = \frac{1}{2k}\{u(x_1 + h, x_2, .., x_k, t - l) - u(x_1 - h, x_2, .., x_k, t - l) + ... + u(x_1, x_2, .., x_k + h, t - l) + u(x_1, x_2, .., x_k - h, t - l)\}$$

⇛ The k dimension problem we can solve in he same way as 1dim.

⇛ In each point we have $2k$ possibility to move(left-right) in each of the dimensions. The probability has to be $\frac{1}{2k}$.

Backup