Differential Evolution MCMC Algorithm

PROJECT – STOCHASTIC MODELLING & SIMULATION

KEERTHANA CHANDRASEKAR JEYANTHI

MARKOV CHAIN MONTE CARLO

- A sampling technique to sample from complicated PDF

e.g. $p(x) = \frac{1}{Z}\tilde{p}(x)$; $Z = \int \tilde{p}(x)\,dx$, where the integral cannot be computed analytically

In a Bayesian problem, the posterior density is given as

$$\pi_n(\theta) \propto \pi(\theta)L(\theta); \qquad \pi(\theta) \text{ -> prior, } L(\theta) \text{ -> likelihood}$$

- MCMC builds an irreducible, aperiodic DTMC (X_t, $t \geq 0$) on χ s.t. $p(x)$ is the stationary distribution equals the posterior distribution of interest.
- Let χ be a continuous space and $p(y|x)$ be a transition kernel s.t. $x, y \in$ χ
- For an ergodic chain, $p(X_1 = y \mid X_0 = x) > 0 \quad \forall x, y \quad as\ \ t \geq 0$ and $\pi(y) = \int \pi(x)p(x|y)dx$ is an invariant measure
- As the chain is ergodic and aperiodic, $\lim_{t\to\infty} p(X_t) = \pi(x)$

METROPOLIS HASTINGS ALGORITHM

To sample from the posterior distribution, Metropolis algorithm uses the prior and the likelihood functions. If the posterior is given as $\pi_n(\theta) \propto \pi(\theta)L(\theta)$

The algorithm is as follows:

- Begin by choosing an initial value θ_0 where the posterior density is positive.
- Proposal = Transition kernel of the Markov Chain, i.e. we propose a new value (θ^*) for $\theta^{(j)}$ which is selected random in the interval $(\theta^{(j)} - C, \theta^{(j)} + C)$, where C is a preselected constant.
- Acceptance condition, we accept the proposed vector according to the Metropolis Hastings acceptance criteria, which accepts the proposal with a probability α given as

$$\alpha = min\left\{1, \frac{\pi_n(\theta^*)}{\pi_n(\theta^{(j)})}\right\}$$

  - Simulate a r.v. Uniform random variable U, if the acceptance probability is greater than U, we accept the proposal θ^* or we stay with the current vector $\theta^{(j)}$
  - The steps are repeated until we generate the required number of samples.

Differential Evolution (DE)

Differential Evolution is a method in evolutionary computation that optimized a problem by iteratively trying to improve a candidate solution with regards to a given measure of quality.

If $f: \mathbb{R}^n \to \mathbb{R}$, is the function to be minimized, a basic algorithm of the Differential Evolution is given as:

- Let a vector $x \in \mathbb{R}^n$, be a candidate solution
- Initialize a population of vectors known as candidate agents with random positions in the search space
- Until a termination criteria, do the following for all the agents x in the population
  - Pick 3 agents a, b, c at random from the population without replacement and they must be distinct from x
  - Compute a new agent position as $y = a + F\,(b - c)$, F is a constant known as the differential weight
  - If $f(y) < f(x)$, then the agent x is replaced with y
- An agent which has the best solution is taken and returned as the best found candidate solution.

Integrating DE with MCMC

- The Differential Evolution can be integrated with MCMC, where multiple chains are run in parallel, where the chains learn from each other.
- This may improve the efficiency of the searching procedure and avoid sampling in the vicinity of a local minimum.
- The new chains are accepted or rejected according to the Metropolis Hastings acceptance criteria
- The algorithm to sample from the posterior $\pi_n(\theta) \propto \pi(\theta)L(\theta)$ is given as follows:
  - Initialize the population $\theta_{i,0}, i \in \{1, \dots, N\}$
  - For each chain θ_i ,the following is done until required number of samples are generated
    - Sample two chains θ_a, θ_b from the population where $\theta_a \neq \theta_b \neq \theta_i$
    - A proposed vector is calculated as $\theta^* = \theta_i + \gamma(\theta_a - \theta_b) + \varepsilon$, where γ is the tuning factor which is set as $\gamma = \frac{2.38}{\sqrt{2d}}$, where d is the dimension of θ and ε is a random value with small variance
    - The proposed vector is accepted with the M-H criteria, i.e with a probability of $\alpha = min\left\{1, \frac{\pi_n(\theta^*)}{\pi_n(\theta_i)}\right\}$
    - If the proposed vector is accepted then, $\theta_i = \theta^*$, otherwise θ^* is unchanged