Application of Monte-Carlo methods to community detection

Diogo Soares, Clément Nicolle, Charles de Haro

February 5, 2022

1 Introduction

1.1 Problem Description

Let's consider the following problem. Let's denote graph G as a random generated graph composed of N nodes $i \in \{1, ..., N\}$ each one belonging to one of two groups. The variable denoting to which group node i belongs is $x_i^* \in \{-1, +1\}^N$. Thus we generate a random vector $x^* \in \{-1, +1\}^N$ with i.i.d elements uniformly distributed in $\{-1, +1\}$, therefore for N large enough, we can see that $\sum_{i=1}^N x_i^* \approx 0$, which means that in expectation the two groups have the same number of elements.

An edge between two nodes i, j in the graph is represented by the variable $e_{ij} \in \{0, 1\}$, where $e_{ij} = 1$ means that this edge exists in the graph. The edges are generated independently as follows:

$$\mathbb{P}(e_{ij} = 1 | x_i^* x_j^* = +1) = \frac{a}{N} = 1 - \mathbb{P}(e_{ij} = 0 | x_i^* x_j^* = +1), \forall i < j \in [|1, N|]$$

$$\mathbb{P}(e_{ij} = 1 | x_i^* x_j^* = -1) = \frac{b}{N} = 1 - \mathbb{P}(e_{ij} = 0 | x_i^* x_j^* = -1), \forall i < j \in [|1, N|]$$

where a, b are two O(1) positive numbers. We assume that $a > b$.

Let's then consider the observation graph $\mathbf{G}$ (or equivalently given the edge-variables e_{ij} for $1 \leq i < j \leq N$), the goal is to find a vector $x \in \{-1, +1\}^N$ as close as possible to x^*.

To assess our estimate x, we define a quantity called *overlap* as

$$q_N = \frac{1}{N} | \sum_{i=1}^N \hat{x_i} x_i^* |$$

The quantity $q_N \in [0, 1]$ and measures the quality of the estimate x. Note that we need the absolute value in the definition of the overlap, since we can only find the communities and not their associated values ± 1. Indeed, both x and -x results in the same distribution for the generated graph.

1.2 Project proposal

In this project we aim to solve the above defined problem, in order to solve the problem we will use Simulated Annealing, more specifically we will try three different techniques:

- Metropolis Hastings.
- Metropolis Hasting with Houdayer move at each step.
- Metropolis Hastings with Houdayer move at for every n steps.

In order to use the Metropolis Hastings algorithm along with its variants we are going to need a function to minimize, as well as a base chain.

Let's consider the energy function to be (in fact the energy can be derived from the posterior distribution)

$$\mathcal{H}(x) = - \sum_{1 \leq i \leq j \leq N} h_{ij} x_i x_j$$

with

$$h_{ij} = \frac{1}{2}\left[e_{ij} \ln \frac{a}{b} + (1 - e_{ij}) \ln \frac{1 - \frac{a}{N}}{1 - \frac{b}{N}} \right]$$

Let's denote the base chain as ψ, the base chain operates on all possible values of $x \in \{-1, +1\}^N$, at each step the base chain will choose uniformly at random a component of x and flip it, therefore we have the following probability matrix.

$$\begin{cases} p_{ij} = \frac{1}{N} & \text{if } \sum_{t=1}^{N} |i_t - j_t| = 1 \\ p_{ij} = 0 & \text{otherwise} \end{cases}$$

As we can see the chain is clearly irreducible, since all states communicate with each other. Additionally, the state space is finite, therefore the chain is positive-recurrent. Even though we won't show the results here, this specific chain, coupled with the above energy function guarantee aperiodicity for the Metropolis Hastings chain, hence ergodicity holds.

2 Theoretical Exercises

2.1 Calculation of r_c

The moment for which the phase transition occurs is defined by the equation

$$(a - b)^2 = 2(a + b)$$

By defining $d = \frac{1}{2}(a + b)$ and $r = \frac{b}{a}$, we can express a and b as functions of d and r :

$$\begin{cases} d = \frac{1}{2}(a + b) \\ r = \frac{b}{a} \end{cases} \iff \begin{cases} a = 2d - b \\ r(2d - b) = b \end{cases} \iff \begin{cases} a = \frac{2d}{1 + r} \\ b = \frac{2rd}{1 + r} \end{cases}$$

By injecting this in the previous equation we obtain :

$$4d^2 \left(\frac{1 - r_c}{1 + r_c} \right)^2 = 4d \iff (d - 1)r_c^2 - 2(d + 1)r_c + d - 1 = 0$$

We recognize a second degree polynomial with discriminant $\Delta = (-2(d + 1))^2 - 4(d - 1)^2 = 16d$ then the solutions are of the form :

$$X = \frac{d + 1 \pm 2\sqrt{d}}{d - 1} = \frac{(\sqrt{d} \pm 1)^2}{(\sqrt{d} - 1)(\sqrt{d} + 1)} = \frac{\sqrt{d} \mp 1}{\sqrt{d} \pm 1}$$

As $r_c \in [0, 1]$ and is a solution of the previous polynomial we finally get :

$$r_c = \frac{\sqrt{d} - 1}{\sqrt{d} + 1}$$

2.2 Ergodicity of Houdayer Move

Let's first define the Houdayer Move.

For Houdayer let's consider x^1 and x^2, such that $x^1, x^2 \in \{-1, +1\}^N$, in order to define the local overlap $\forall i : y_i = x_i^1 * x_i^2$.

Then for every move, we selected the indices with $y_i = -1$.

Afterwards, we create the graph G', which is a copy of G with only the nodes i where $y_i = -1$.

Furthermore, we select uniformly at random a node t from G' and all the nodes reachable from t through the graph edges, we denote this to be node t cluster.

In the end, we flip all the nodes belonging to node t cluster in the configurations x^1 and x^2.

Let's now look at the Ergodicity of this move.

Claim: y is invariant under the Houdayer Move.

Proof: Let's consider an houdayer move on node t, which belongs to cluster C_1, then after the Houdayer move we get,

$$\begin{cases} y_i = (-x_i^1) * (-x_i^2) & \text{if } t \in C_1 \\ y_i = x_i^1 * x_i^2 & \text{otherwise} \end{cases}$$

Clearly in both scenario y_i remains the same, hence completing the proof.

It's clear that the move is reducible, for example let's consider a configuration (x^1,x^2), such that we have y' different than y, it's quite obvious that this configuration will never be reached, since y is invariant under the Houdayer move. Therefore concluding that the houdayer move is reducible.

Since y is invariant under the Houdayer move, then G' is also invariant under the Houdayer move. Furthermore, we can define $C_1, C_2, ..., C_T$ as the clusters of G', and these clusters will be invariant as well.

Let's now define the Houdayer move with a Markov chain, since we can define each configuration (x^1, x^2) by whether each cluster is flipped or not let's consider each state as $S \in \{0,1\}^T$, with the ith entry corresponding to whether C_i is flipped or not . It's clear that since we are picking uniformly at random then the $\mathbb{P}(t \in C_i) = \frac{\#C_i}{N}$, therefore we define the following probability matrix:

$$\begin{cases} p_{ij} = \frac{\#C_k}{N} & \text{if } \sum_{t=1}^{N} |i_t - j_t| = 1, \text{ for } t \text{ s.t } i_t \neq j_t \text{ let } t \in C_k \\ p_{ij} = 0 & \text{otherwise} \end{cases}$$

We can notice that $\forall i \in Sp_i^{2k} > 0, k \in \mathbb{N}$, conversely $\forall i \in Sp_i^{2k+1} = 0, k \in \mathbb{N}$ and $\forall i \in Sp_i^0 = 0$, which means the Markov Chain is periodic with period 2.

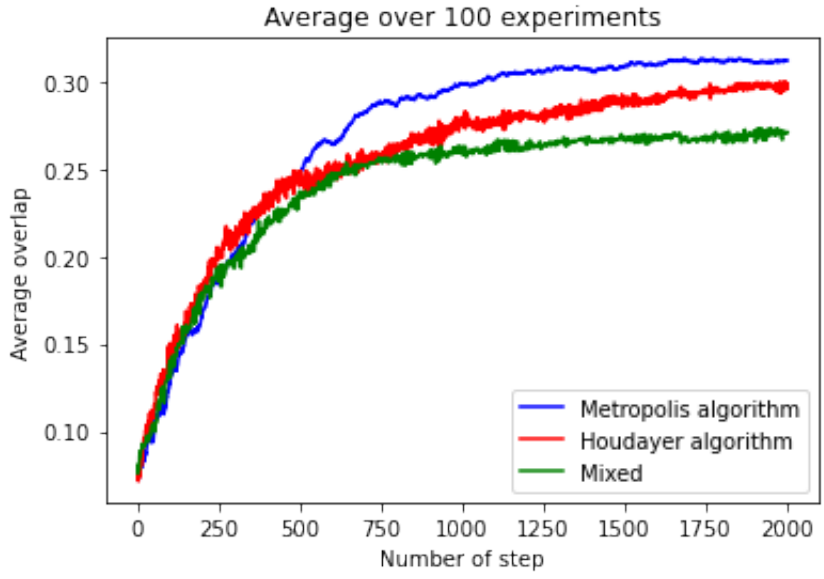
We have seen in class that only a positive-recurrent, irreducible and aperiodic Markov Chain can be ergodic, which means that the Houdayer move is not ergodic since it's not aperiodic, nor irreducible.

3 Exploration Results

3.1 Average Overlap / Steps

The parameters for this study are

- N = 100 , d = 3 and a = 5,5
- 2000 steps over 100 experiments
- initial beta = 0.01 with linear update of 0.01
- in the mixed algorithm we do an Houdayer move every 5 steps

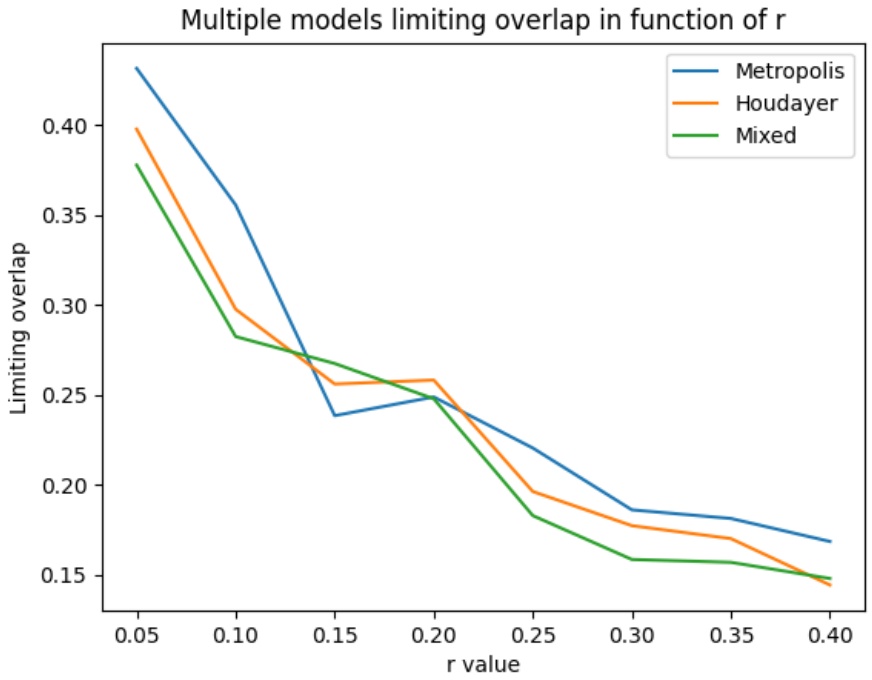


This graphic highlights that the Houdayer algorithm is the slowest to converge (continue to increase while both of the Metropolis and Mixed algorithms curves are almost flat since the 1500th step)

3.2 Limiting Overlap / R

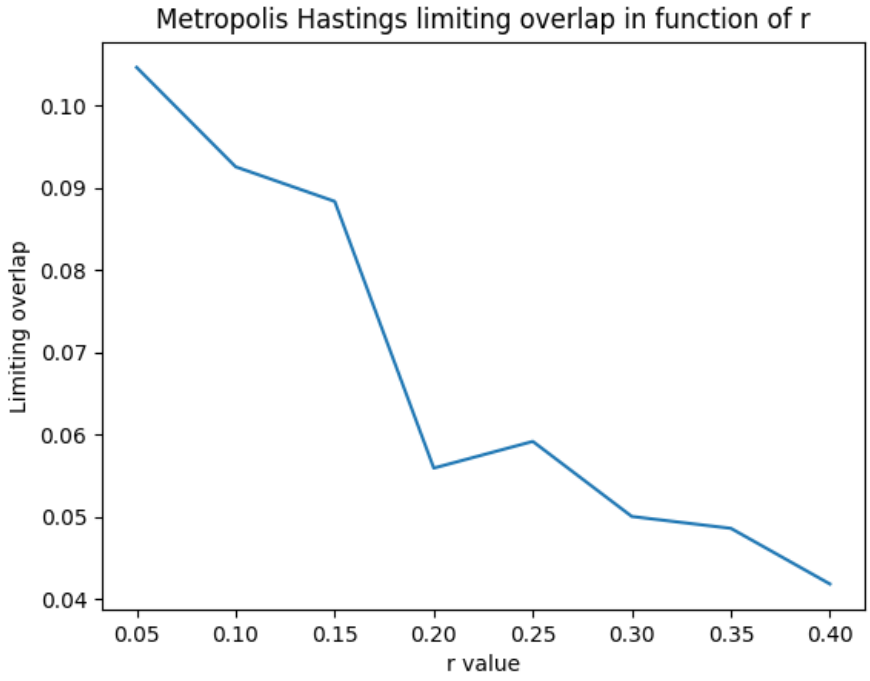
Let's set the parameters of our test:

- N = 100
- d = 3
- 1000 steps
- 100 experiments
- initial beta = 1 with linear update of 0.01
- in the mixed algorithm we do an Houdayer move every 5 steps



As we can see in most r values, the Metropolis algorithm provides the best results.

Moreover we also tested the metropolis hastings algorithm for a bigger N, let's now consider N = 1000, with 6000 steps



Even though the nominal results for the limiting overlap are different, we see that the functions behave in a similar way.

3.3 Phase Transition

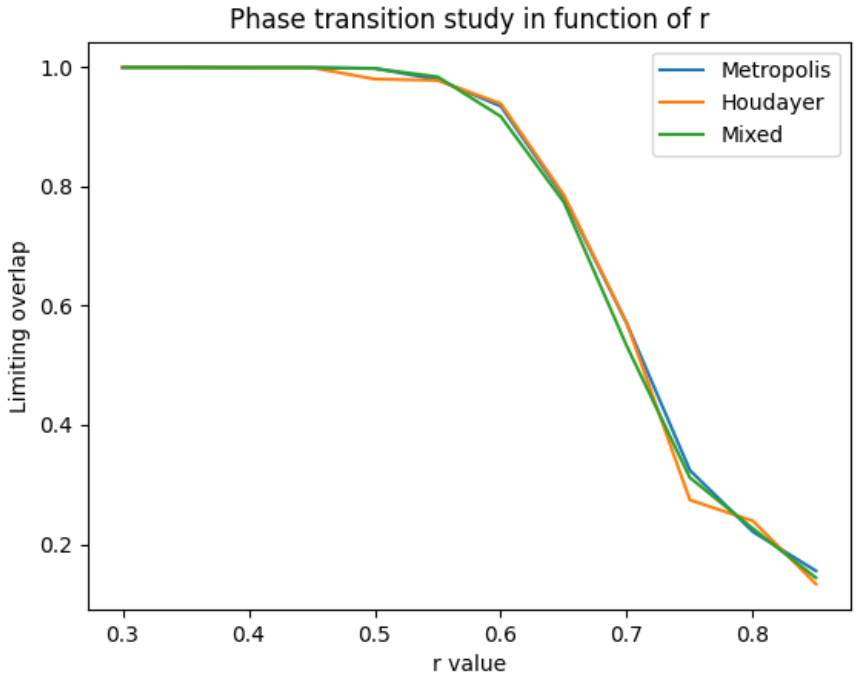
Let's set the parameters of our test:

- N = 100
- d = 44
- 1000 steps
- 100 experiments
- initial beta = 1 with linear update of 0.01
- in the mixed algorithm we do an Houdayer move every 5 steps

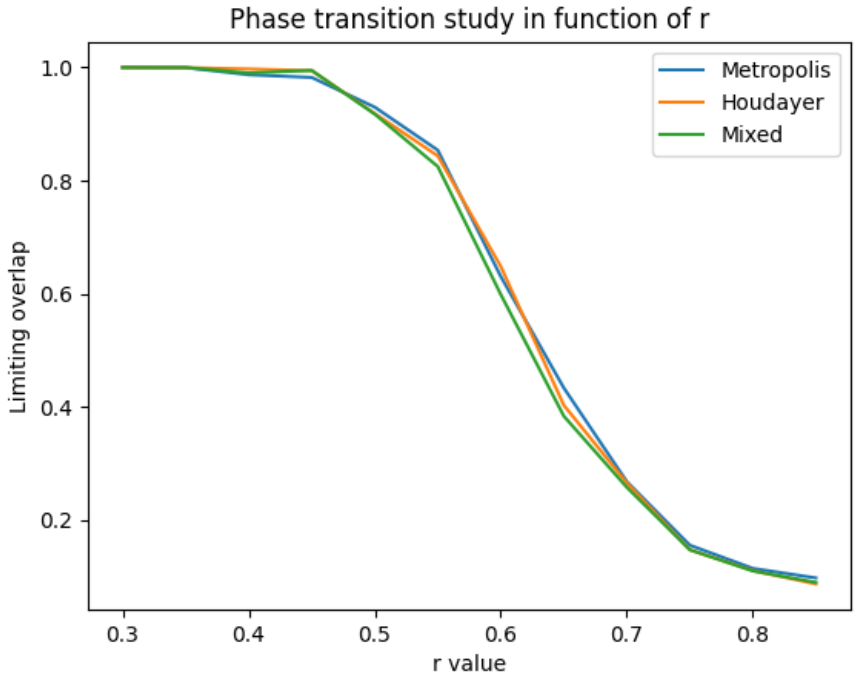
We consider a bigger d in order for the phase transition to be more visible, from the theoretical results we compute the phase transition to be

$$r_c = \frac{\sqrt{44}-1}{\sqrt{44}+1} = 0.738$$

.



Moreover we also tested the results for a bigger N, let's now consider N = 200. The phase transition should be the same since it doesn't depend on N.



It's interesting to note that for smaller nominal differences, between $\frac{a}{N}$ and $\frac{b}{N}$ the phase transition is more slower, however as we can see the phase transition threshold is invariant under N.

4 Conclusion

Throughout our experiments, we tested the performance and the results of different algorithms on the Stochastic Block Model.

In the first experiment, we noticed that in the long run the metropolis algorithm usually provides better results, however it's interesting to see that in the beginning the mixed algorithm outperforms both Metropolis and Houdayer.

In one of the theoretical question we pointed out the fact that the Houdayer move is not ergodic, that being said we expected the Houdayer algorithm to have different limiting results compared to the classic Metropolis algorithm, and our experiment in section 3.2 successfully showed this.

In our second experiment, we noticed that the Metropolis algorithm provided the best results, one possible explanation relates to the consequences of the Houdayer move in this specific problem, in the original paper the Houdayer Move does not change the joint energy of the two configurations, therefore helping the optimization procedure, since it provides a walk on high energy states. Conversely in our problem the energy changes, more specifically, empirically we noticed that for high energy states the Houdayer move would often mean a energy decrease, which in turn lead to worse limiting results.

Finally, in the third experiment we tested the correctness of our theoretical results for the phase transition, fortunately all the algorithms showed the same curve for the analysis of the phase transition. We spotted that the transition point is exactly the one found with the theoretical expression, therefore assuring the theoretical derivation.

To conclude, we think that the Houdayer move does not improve the performance nor the mixing times, even if we consider the mixed Metropolis-Houdayer algorithm. The only benefit we see in the Houdayer move is speeding up the initial part of the Metropolis algorithm at the beginning of the execution.