Random Samplings Using Metropolis Hastings Algorithm

Miguel Arcos-Argudo(✉), Rodolfo Bojorque-Chasi, and Andrea Plaza-Cordero

Research Group on Artificial Intelligence and Assistance Technologies (GIIATA), Salesian Polytechnic University, Cuenca, Ecuador
{marcos,rbojorque,aplaza}@ups.edu.ec

Abstract. Random Walks Samplings are important method to analyze any kind of network; it allows knowing the network's state any time, independently of the node from which the random walk starts. In this work, we have implemented a random walk of this type on a Markov Chain Network through Metropolis-Hastings Random Walks algorithm. This algorithm is an efficient method of sampling because it ensures that all nodes can be sampled with a uniform probability. We have determinate the required number of rounds of a random walk to ensuring the steady state of the network system. We concluded that, to determinate the correct number of rounds with which the system will find the steady state it is necessary start the random walk from different nodes, selected analytically, especially looking for nodes that may have random walks critics.

Keywords: Markov chains · Small worlds · Metropolis hastings · Random walks · Node sampling · Random sampling

1 Introduction

Actually, most complex systems such as biological cell development networks and brain activity are studied under their network structure [1] to understand how it communicate, work, self-organize, evolve, etc. This allows to understand the importance of the subject of our study within the field of computing, since the application of random sampling can be seen in various environments such as for the efficient collection of energy data in wireless sensor networks [2], analysis of social networks and information [3], operations on big data [4], etc. This paper presents the results of random walks in a graph with characteristics of a Markov Chain, by implementing the Metropolis-Hastings Random Walks (MHRW) algorithm, with which random samples can be obtained after a certain number of jumps (rounds), and in such a way that all the nodes of the network have a uniform probability of being chosen as a sample. The work has determined the number of rounds with which a uniform sampling distribution is obtained, and with a stable average error which is called the "steady state of the network", regardless of the node by which the walk begins. In addition, experiments have been carried out starting the random walks from initial nodes chosen in an analytical way, taking into account the main concepts and metrics on graphs, which has allowed executing a possible method of finding nodes that present special characteristics and provoke that the random path is come back critical.

2 MHRW Algorithm

The importance of the implementation of sampling algorithms and random walks on networks is studied by Sevilla et al. [5], Arcos [6], among others. According to [7] the Metropolis-Hastings Algorithm (M–H) is extremely versatile, and is widely used in physics, the authors use (1) as probability density function:

$$\int q(x, y)dy = 1 \tag{1}$$

This density is capable of generating a new state y when the current state is x starting from $q(x, y)$, however (1) does not guarantee that the probability of staying in the state x is the same as passing to the state y (condition of reversibility), since the probability of transiting from x to y would be greater. Then, it is convenient to include a condition that reduces the number of state changes from x to y, by entering a probability $\alpha(x, y) < 1$; if a state change does not occur, it remains in x returning this value for the given distribution. Then, we can establish the following relationship:

$$p_{MH}(x, y) \equiv q(x, y)\alpha(x, y) \,\text{for}\, x \neq \text{y} \tag{2}$$

where $\alpha(x, y)$ is still to be determined. With $PMH(x, y)$ he reversibility condition can be satisfied as shown below (consider that $\alpha(y, x) \approx 1$):

$$\begin{aligned} \pi(x)q(x, y)\alpha(x, y) &= \pi(y)q(y, x)\alpha(y, x) \\ &= \pi(y)q(y, x) \end{aligned} \tag{3}$$

where π represents the desired distribution, then:

$$\alpha(x, y) = \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)} \tag{4}$$

Thus, the authors conclude that $PMH(x, y)$ is reversible. The probability of obtaining a new value of the state during the tour is:

$$\alpha(x, y) = \begin{cases} min\left[\frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}, 1\right], \; if \; \pi(x)q(x, y) > 0 \\ \qquad 1, \; any \; other \; case \end{cases} \tag{5}$$

3 Concepts and Basic Notions

A network of any nature can be represented by a graph which consists of a finite set of nodes. Each node represents an element of the network and must have a unique identifier; the way in which network elements are connected are represented by edges; an edge is a line joining a pair of nodes; when a node O is directly connected to a node P by an edge, we can say that O and P are neighboring nodes. In a directed graph each edge has an address, that is, each edge a_i that begins in a source node O allows jump to a destination node P, but it is impossible to jump from node P to node O using the same edge a_i, Fig. 1. In an undirected graph edges are bidirectional, that is, it is possible to jump from node O to node P, and form node P to node O using the same edge a_i for both cases, Fig. 2. In an undirected graph the number of edges that are connected to it determines the degree of a node. A connected graph is a graph in which all its nodes are connected to each other by edges (Fig. 1). A disconnected graph is a graph in which a node or group of nodes in the graph are isolated from another group of nodes (Fig. 3). A weighted graph is a graph in which each a_i has an associated weight w_i, this weight represents the cost of jumping from the origin node O to the destination node P using the edge a_i (Fig. 4). An unweighted graph is a graph in which all its edges have the same associated weight, in this case the weight of each edge a_i in the graph is not graphically represented (Fig. 2). The present study uses an undirected, connected and unweighted graph, which coincides with the representation of Fig. 2.

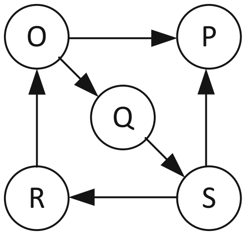

Fig. 1. Directed, connected and unweighted graph.

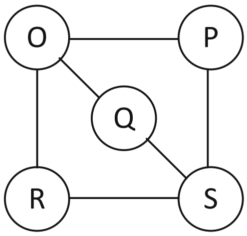

Fig. 2. Undirected, connected and unweighted graph.

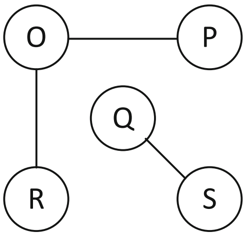

Fig. 3. Undirected, disconnected and unweighted graph.

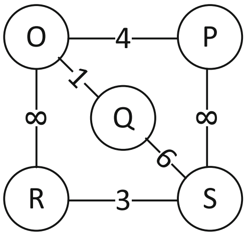

Fig. 4. Undirected, disconnected and unweighted.

In a random walk we call the node in which the state of the network is located at an instant of time *t*, that is, if the state of the network at time *t* is in node *O*, then, the node *O* is the current node at time *t*, if the next state of the network at time *t* + *1* is the node *P*, then node *P* is the current node at time *t* + *1*.

A random walk is constructed in the following way: given a graph *G* and an initial node *O* as starting point (current node) in which the state of the network is located at time *t* = *0*, a neighbor node of *O* is randomly selected, this can be the node *P*, then, the state of the network passes to node *P* at time *t* = *1*, then *P* is the current node, then, a neighbor node of *P* is selected randomly, this can be the node *Q*, and the state of the network passes to node *Q* which is the current node at time *t* = *2*, this is repeated successively until a stop condition is reached. It should be noted that in order to determine that a network passes from one state to another, some condition may be imposed, in case this condition is not fulfilled, and the node *O* is the current node, the network would not change state and the current node would continue being the node *O* [8–10], in this work this condition is defined in (6).

Our goal is to demonstrate that by using the MHRW algorithm we can perform a uniform sampling of the nodes of a connected, undirected and unweighted graph, and stabilize the mean error with respect to the uniform distribution, understood by uniform distribution to all the nodes have the same probability of being chosen as a sample.

4 Discussion

We have used a graph with characteristics of Markov chain that was obtained from the Network Data Repository[1], it represents the Facebook network of the Massachusetts Institute of Technology (MIT), it is composed by 6402 interconnected nodes through 251230 edges. For each experiment, a number of samples equivalent to 100 * number of graph nodes was obtained uniformly. In the simulation, 100 random walks were executed by randomly selecting the initial node, and also by taking at convenience nodes that present relevant characteristics in graph analysis; the results are summarized in Table 1. The condition of the implementation with which it is satisfied (5), is given by the generation of a random value p in a uniform manner, such that:

$$p \leq \min\left[\frac{degree(currentnode)}{degree(neighborNode)}, 1\right] \tag{6}$$

That is, the higher the degree of the neighbor selected at random, the less likely it is that it will be visited or chosen as a sample (current node); when the condition of (6) is met, the current node becomes the selected neighbor. The criterion to stop the execution of the simulation in each case was given by the stabilization of the mean error calculated by (7), that is, when the value of the error showed no significant changes, the experiment was stopped.

$$E = \frac{\sum_{i=1}^{n} \frac{m_i - freq}{freq}}{n} \tag{7}$$

Where n is the number of nodes in the graph, m_i is the number of times the node i was taken as a sample, and *freq* is the sampling frequency. For the random selection of the neighbor, a uniform distribution function was used [11]; it was possible to verify that this function generates random numbers with a distribution similar to the distribution of the random selection of the neighboring nodes during all the samplings made in all experiments (Figs. 5 and 6).

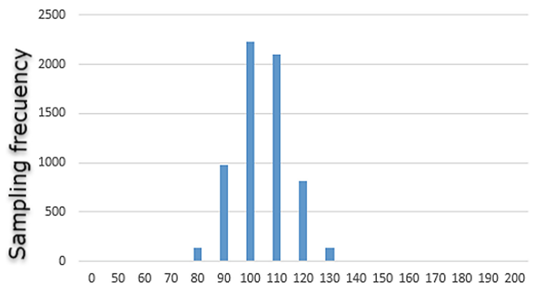

Fig. 5. Distribución del muestreo de los nodos seleccionados después de paseos de 20.000 rondas.

[1] http://networkrepository.com/.

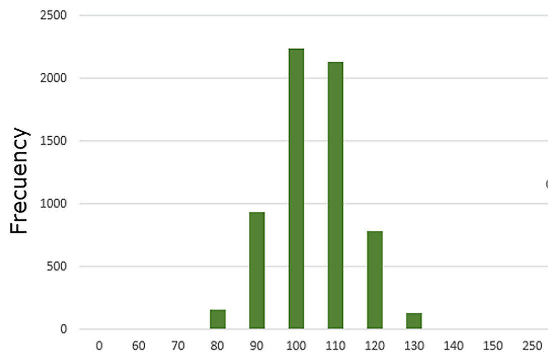

Fig. 6. Distribution of random numbers generated with the uniform distribution function.

Analyzing the results obtained, we conclude that, for the set of tests proposed, the average error is stabilized at approximately 8%, as shown in Table 1, which is corroborated by making increasingly long random walks, we have conducted experiments with 20,000 rounds prior to taking the sample. The variation of the average error can be seen in Fig. 7. The first column of Table 1 shows the identifier of the node, the second column shows the name of the metric by which the node was chosen, the third column shows the value of the metric specified in the second column, the fourth column shows the number of rounds needed for the average error value to reach a steady state, the fifth column shows the value of the average error obtained during the random walk ($\approx$ 8% in all the cases). As can be seen in Table 1, when the random walk was initiated by the node 1010 whose eigenvector value is the highest of all, the number of necessary rounds increased considerably compared to the previous nodes, this led us to analyze the characteristics of this node; it is a node of degree 1 whose only neighbor had a significantly higher degree, later we verified if there was another node whose degree is 1 and whose only neighbor has an even greater degree than the previous one, this is how we find the node 4503.

Table 1. Results of executed experiments

Node	Metric	Value	Rounds	Avg. error
2411	Random selected node	–	400	0.081
6400	Lowest degree	1	470	0.081
2982	Highest degree	708	350	0.080
30	Lowest betweenness	0	750	0.081
1940	Highest betweenness	4.9E + 12	300	0.083
173	Lowest coefficient clustering	0	400	0.083
3302	Highest coefficient clustering	1	500	0.080

(*continued*)

Table 1. (*continued*)

Node	Metric	Value	Rounds	Avg. error
3277	Lowest closeness	0.181	500	0.080
2982	Highest closeness	0.495	300	0.083
558	Lowest eigenvector	0.001	290	0.083
1010	Highest eigenvector	4,9+12	1600	0.079
4503	Critic node	–	6000	0.078
	Average error			0.081

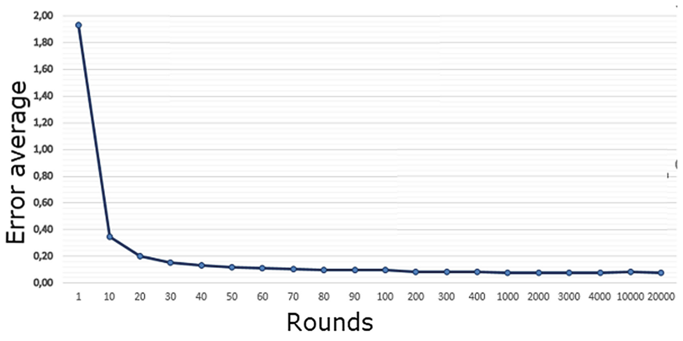

Fig. 7. Variation of the average error of the random walk initiated by a node chosen randomly (Node 2411).

When the random walk was started by this node, the value of the average error decreased at a significantly lower speed, so this became the most critical route of the set of tests performed, the results are shown in the penultimate row of Table 1. The feature of node 4503 is that its degree is 1, and its only neighbor (node 2720) has a degree of 617, so meeting a condition that satisfies Eq. (6) has a probability of 1/617, then, the random walks turned out to be longer. Sampling reached a steady state after a walk of approximately 6,000 rounds. That is why we have verified that the number of rounds with which it is guaranteed that the random walks in the graph arrive at a stationary state applying MHRW is 6,000, regardless of the node by which the route begins, which coincides with the result published in [5].

In Table 2 the first column shows the number of rounds used in each random walk, the second column indicates the value of the average error obtained in each trip, the third column shows the identifier of the most sampled node in each random walk, the fourth column shows the number of times that the most sampled node was selected as a sample; In this table it can be seen that in random walks that do not reach 6000 rounds the most sampled node is 4503, that is, due to the small probability of jumping to its neighbor node, the most sampled node is the same node, which prevents reaching the steady state. Note also that although at 4000 rounds the mean average error has decreased to a value of 0.081, the number of times that node 4503 has been sampled is approximately ten times greater than the desired average, so the uniform distribution

Sampling is achieved only in a walk of 6000 rounds, under these conditions the most sampled node is a node different from the initial node. Then, a possible method to find initial nodes that present critical random walks could be taking as initial node the node whose value of eigenvector is the highest of all.

Table 2. Nodes most sampled in random walks that started by node 4503 according to the number of rounds.

Rounds	Avg. error	Most sampled node	Sampling frequency
100	2	4503	545013
200	1	4503	463663
300	1	4503	394263
400	1	4503	334579
500	0.891	4503	284823
600	0.759	4503	242464
700	0.645	4503	206069
800	0.549	4503	175609
900	0.469	4503	149929
1000	0.398	4503	126889
1100	0.341	4503	107955
1200	0.291	4503	91417
1300	0.254	4503	78415
1400	0.221	4503	66384
1500	0.196	4503	56738
1800	0.144	4503	34783
2100	0.117	4503	21413
3000	0.088	4503	5062
4000	0.081	4503	1054
5000	0.081	4503	291
6000	0.078	3267	138

5 Conclusions

The MHRW algorithm guarantees that the walk in a graph are completely random, giving all the nodes the same probability of being visited and taken as a sample, so it is an important tool when performing statistical and predictive analyzes; it should be considered that the algorithm in its implementation requires a random number generator engine whose distribution directly influences the mean error value of the sampling. From the results obtained it can be concluded that it is important to test the random walks starting from different nodes, but not only randomly, is necessary choosing them analytically considering the characteristics of the graph, then, is possible deduce ways of looking for other nodes that are interesting for the analysis and to determine a number of rounds with greater precision. Finally, we conclude that the value of the mean error obtained in a sample will be smaller the larger the sample size obtained.

Random walks should be carried out by increasing the number of rounds progressively, in such a way that a sufficient number of rounds can be established to guarantee the steady state of the network, but that it does not generate unnecessary computational cost.

References

1. Brugere, I., Gallagher, B., Berger-Wolf, T.Y.: Network Structure Inference, A Survey: Motivations, Methods, and Applications. arXiv preprint arXiv:1610.00782 (2016)
2. Baqer, M., Al Mutawah, K.: Random node sampling for energy efficient data collection in wireless sensor networks. In: 2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp. 467–472. IEEE (2013)
3. Blagus, N., Weiss, G., Šubelj, L.: Sampling node group structure of social and information networks. arXiv preprint arXiv:1405.3093 (2014)
4. Gadepally, V., Herr, T., Johnson, L., Milechin, L., Milosavljevic, M., Miller, B.A.: Sampling operations on big data. In: 2015 49th Asilomar Conference on Signals, Systems and Computers, pp. 1515–1519. IEEE (2015)
5. Sevilla, A., Mozo, A., Anta, A.F.: Node sampling using random centrifugal walks. J. Comput. Sci. **11**, 34–45 (2015)
6. Arcos Argudo, M.: Comparative study between Kleinberg algorithm and biased selection algorithm for construction of small world networks. Computación y Sistemas **21**(2), 325–336 (2017)
7. Chib, S., Greenberg, E.: Understanding the Metropolis-Hastings algorithm. Am. Stat. **49**(4), 327–335 (1995)
8. Lovász, L.: Random walks on graphs. Combinatorics, Paul erdos is eighty **2**, 1–46 (1993)
9. Aldous, D., Fill, J.: Reversible Markov Chains and Random Walks on Graphs (2002)
10. Woess, W.: Random Walks on Infinite Graphs and Groups, vol. 138. Cambridge University Press, Cambridge (2000)
11. Freund, J.E., Miller, I., Miller, M.: Estadística matemática con aplicaciones. Pearson Educación (2000)