Exercise 03 Sthavishtha Bhopalam Rajakumar December 28, 2017

1 Introduction

The objective of the current work is to implement Hoshen-Kopelmann (HK) algorithm to plot the cluster size distribution for varying occupation probabilities. Advantages of this algorithm over the Burning algorithm include less memory requirements and computational power. Based on the *union-finding* algorithm [1], this finds its application in the cutting-edge research domain - Machine Learning and Data Mining.

2 Algorithm Description

Some important points in implementing the HK algorithm include:

- Let each lattice site be designated $N(i, j)$ which has a value between $0, 1$, where 0 refers to an empty lattice and 1 to an occupied one. Additionally, k denotes the cluster number.
- Initially, k starts from 2 (as the cluster numbers $0, 1$ are already used). While moving through each lattice site, if there is an occupied one, we add $M(k)$ to 1 and $N(i, j)$ to k.
- The remaining steps are summarized in the algorithm shown below.

To summarize, the algorithm of HK can be given as follows [2]:

for $i, j \leq N$ **do**
    $N(i, j) \leftarrow 0, 1 from\ drand48()$
end for
for $i, j \leq N$ **do**
    $N(first\ occupied\ site) \leftarrow k$
    $M(k) \leftarrow 1$
end for
if top and left sites are empty **then**
    $k \leftarrow k\ +\ 1$
    $N(i, j) \leftarrow k$
    $M(k) \leftarrow 1$
else
    if One is occupied with k_o **then**
        $N(i, j) \leftarrow k_o$
        $M(k) \leftarrow M(k)\ +\ 1$
    else
        if Both are occupied with k_1 **then**
            $N(i, j) \leftarrow k_1$
            $M(k) \leftarrow M(k)\ +\ 1$
        else
            Choose a smaller one among $k_1\ or\ k_2$ (say k_1 for instance)
            $N(i, j) \leftarrow k_1$
            $M(k_1) \leftarrow M(k_1)\ +\ M(k_2)\ +\ 1$
            $M(k_2) \leftarrow -k_1$
        end if
        In the above steps, if any $M(k)$ happens to be negative, then the original referencing cluster has to be found as :
        while $M(k) \leq 0$ **do**
            $k \leftarrow -M(k)$

end while
end if
end if
for $k = 2; k \leq k_{max}$ **do**
if $M(k) \geq 0$ **then**
$n(M(k)) \leftarrow n(M(k)) + 1$
end if
end for

3 Results and Discussions

The importance of cluster distribution lies in the fact that different values of occupation probabilities (when compared to the critical occupation probability) result in different results (namely, sub-critical, over-critical and critical regimes). As seen in the Figure 1, the cluster distribution for sub-critical regime (with $p \leq p_c$) varies in accordance with a power law multiplied with an exponential function. Additionally, at critical regime (with $p = p_c$ taken to be 0.592746), a linear behavior between the cluster size and occupation probability can be observed yielding an actual slope of -1.92. This is consistent with the previously determined slopes of 187/91, implying the present results to be credible. For an improved accuracy, it is advisable to use a higher number of lattice sites [1] (of the order 4000×4000 [3]), with more sample runs. However, at over-critical regimes (with $p \geq p_c$), the cluster size distribution resembles an exponential decay.

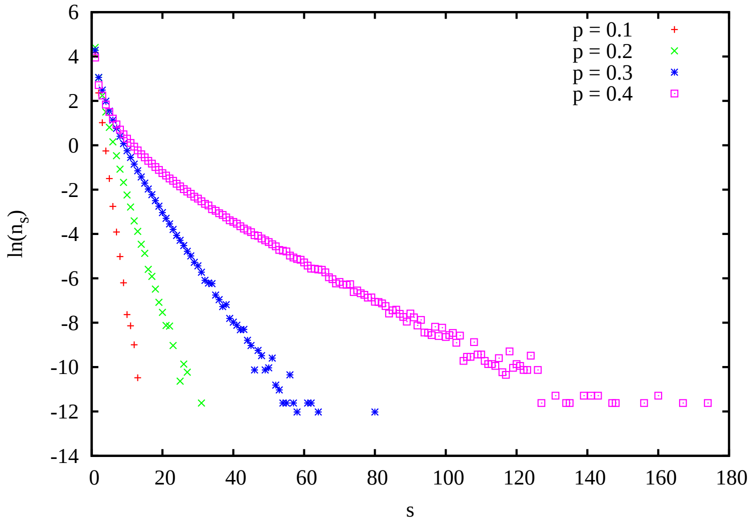
References

[1] Wikipedia contributors. Hoshenkopelman algorithm — wikipedia, the free encyclopedia, 2017. [Online; accessed 27-December-2017].

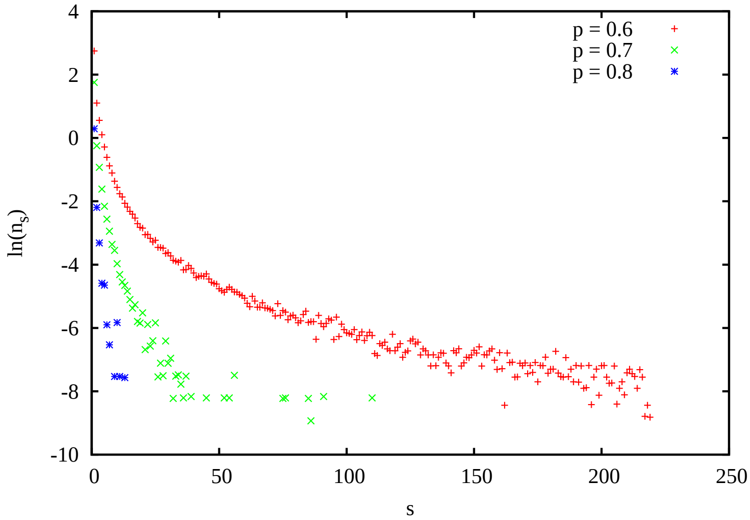
[2] Hans Hermann. Lecture notes on introduction to computational physics. Institute for Building Materials, ETH Zürich, 2017.

[3] J Hoshen, D Stauffer, G H Bishop, R J Harrison, and G D Quinn. Monte carlo experiments on cluster size distribution in percolation. *Journal of Physics A: Mathematical and General*, 12(8):1285, 1979.

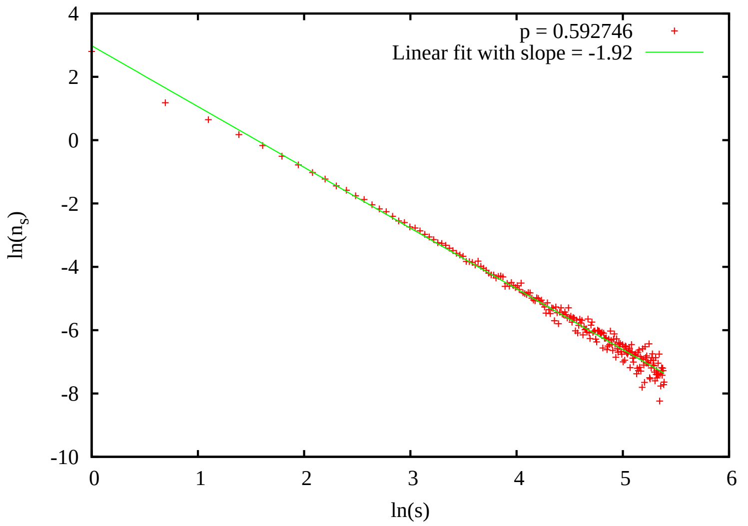
[1]For the current work, 500×500 lattice sites with 1000 runs were used to capture the results.



(a) Sub-critical regions , $p \leq p_c$



(b) Over-critical regions , $p \geq p_c$



(c) Critical regime , $p \ = \ p_c$

Figure 1: Cluster size distributions using HK algorithm