# 20 Probability

## 20.1 The Percolation Model

## **Question 1:**

(a) Why $\theta$ is non-decreasing in $p$?

According to the notes at the end of the project, it suggests that we may label each edge with a pseudo-random number which is uniformly distributed from 0 to 1 and use the same realization for all values of $p$ simultaneously. We assume an edge is open when the assigned number of the edge is $\leq p$.

Let $\mu(\mathrm{p}_1)$ and $\mu(\mathrm{p}_2)$ be the set of edges that are open under the $p = \mathrm{p}_1$ and $p = \mathrm{p}_2$,
Under the same realization, we can deduce that $|\mu(\mathrm{p}_1)| \leq |\mu(\mathrm{p}_2)|$ if $\mathrm{p}_1 \leq \mathrm{p}_2$ $(*)$
As there are more edges available for $p = \mathrm{p}_2$ than for $p = \mathrm{p}_1$ (for which $\mu(\mathrm{p}_1) \subseteq \mu(\mathrm{p}_2)$), we can conclude that
$\{\boldsymbol{y}: \boldsymbol{y}\ are\ accessible\ nodes\ as\ p = \mathrm{p}_1\} \subseteq \{\boldsymbol{y}: \boldsymbol{y}\ are\ accessible\ nodes\ as\ p = \mathrm{p}_2\}$.
Therefore, $\theta(\mathrm{p}_1) \leq \theta(\mathrm{p}_2)$ if $\mathrm{p}_1 \leq \mathrm{p}_2$.

(b) Show $\theta_\mathrm{n}(\mathrm{p})$ is decreasing in $\mathrm{n}$.

For $\theta_\mathrm{n}(\mathrm{p}) = P_p(|C_n| \neq \emptyset)$, we can see it is dependent on $\theta_{\mathrm{n}-1}(\mathrm{p}) = P_p(|C_{n-1}| \neq \emptyset)$ as we know if $|C_{n-1}| = \emptyset$, $|C_n| = \emptyset$ must be true as well.

Therefore,

$$
\begin{aligned}
&\theta_\mathrm{n}(\mathrm{p}) = P_p(|C_n| \neq \emptyset) \\
&= P_p\left(|C_n| \neq \emptyset \bigcap |C_{n-1}| \neq \emptyset\right) \\
&= P_p((|C_n| \neq \emptyset)|) \mid (|C_{n-1}| \neq \emptyset\ ))P_p(|C_{n-1}| \neq \emptyset\ ) \\
&\leq P_p(|C_{n-1}| \neq \emptyset\ ) \\
&\leq \theta_{\mathrm{n}-1}(\mathrm{p})
\end{aligned}
$$

$\blacksquare$

(c) Give an estimate for the likely size of the error $\hat{\theta}_{m,n}(p) - \theta_{\mathrm{n}}(p)$.

We use Central Limit Theorem in this question.

Central Limit Theorem stated that:

$'If\ X_1, X_2, \dots., X_n\ are\ i.i.d. random\ variables\ having\ the\ same\ distribution$

$with\ mean\ \mu\ and\ variance\ \sigma^2.$

$Then\ if\ n \to \infty, the\ random\ variable$

$$Z = \frac{\sum_{i=1}^{n} X_i - n\mu}{\sigma\sqrt{n}}$$

$has\ the\ standardistribution\ d\ normal\ N(0,1)'.$

---

$$\hat{\theta}_{m,n}(p) - \theta_{\mathrm{n}}(p) = \frac{1}{m}\sum_{j=1}^{m} I_n(j) - \theta_{\mathrm{n}}(p)$$

I suggest $\mathrm{I_n}(j)$ has a binomial distribution with probability $P(\mathrm{I_n}(j) = 1) = \theta_{\mathrm{n}}(p)$ and $P(\mathrm{I_n}(j) = 0) = \left(1 - \theta_{\mathrm{n}}(p)\right)\ and\ variance\ \sigma^2 = \left(1 - \theta_{\mathrm{n}}(p)\right)\theta_{\mathrm{n}}(p).$

$\mathrm{Therefore, we\ get}$

$$\frac{\sqrt{m}}{\sqrt{\left(1 - \theta_{\mathrm{n}}(p)\right)\theta_{\mathrm{n}}(p)}}\left(\frac{1}{m}\sum_{j=1}^{m} I_n(j) - \theta_{\mathrm{n}}(p)\right) \sim N(0,1)\ by\ Central\ Limit\ Theorem.$$

Hence,

$$P\left(\left|\frac{\sqrt{m}}{\sigma}\left(\frac{1}{m}\sum_{j=1}^{m} I_n(j) - \theta_{\mathrm{n}}(p)\right)\right| \le \varepsilon\right) = \Phi(\varepsilon) - \Phi(-\varepsilon)$$

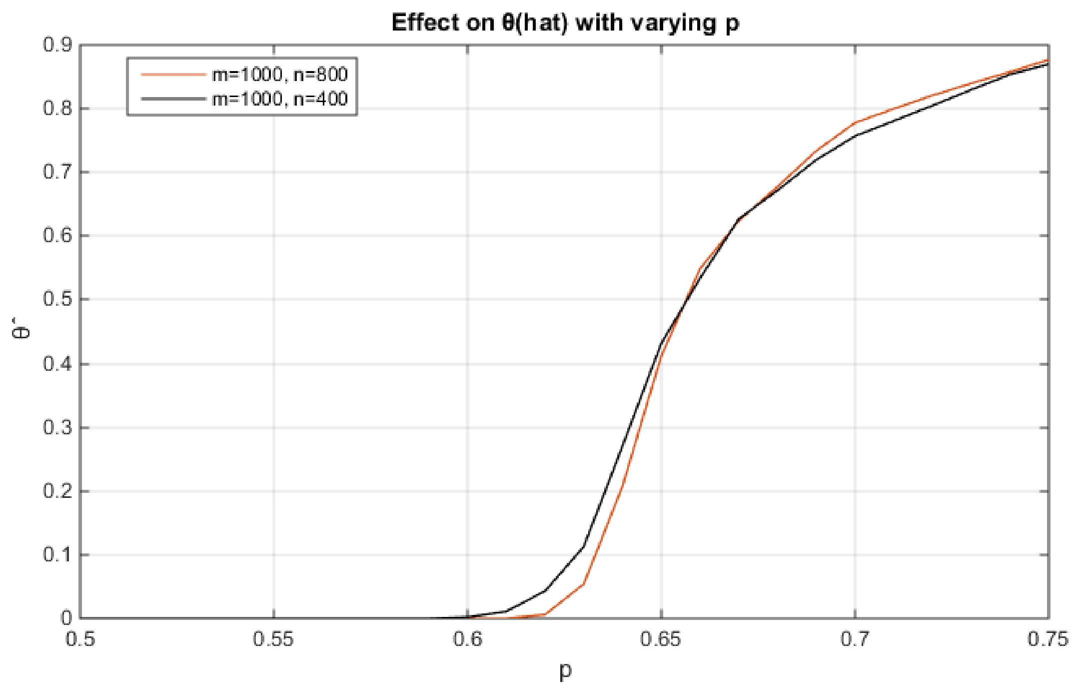
$$P\left(\left|\left(\frac{1}{m}\sum_{j=1}^{m} I_n(j) - \theta_{\mathrm{n}}(p)\right)\right| \le \frac{\sigma}{\sqrt{m}}\varepsilon\right) = \Phi(\varepsilon) - \Phi(-\varepsilon)$$

For a fixed $\varepsilon\ and\ \sigma$, we can reduce the error of $\hat{\theta}_{m,n}(p) - \theta_{\mathrm{n}}(p)$ by increase m. The order of the reduction in error will be in order $O(\frac{1}{\sqrt{m}})$

■

## Question 2:

Regarding to the computational time of the program, I have chosen (m=1000, n=800) and (m=1000, n=400) for my program [percolationQ2_mod.m] to plot $\hat{\theta}_{m,n}(p)$ against $p$ for $p \in [\,0.5\ ,0.75\,]$, which takes around 41 seconds to run for $p_c \approx 0.62$, which is close to the our believed result, $0.644$.

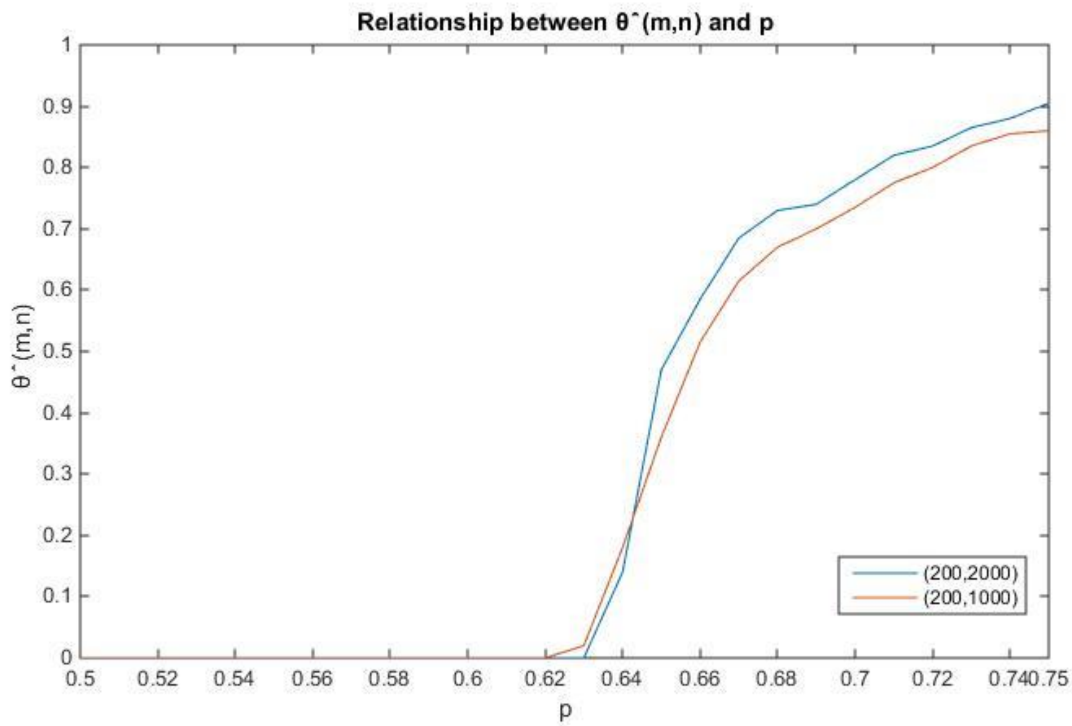



Graph 1: Relationship between $\hat{\theta}$ $and\ varying\ p$

In my program [percolationQ2_mod.m], I have first generate 2 random matrix $A_{ij}$ $and$ $B_{ij}$ of size $(n * n)$ representing the values on the vertical and horizontal edges from node $(i-1, j-1)$. It runs at $O(n^2)$. Secondly, I followed the suggested algorithm from the notes and calculate $Z(\boldsymbol{x})$ for all nodes $\boldsymbol{x}$ on the grid which involves 3 comparison operations per node which is running at $O(n^2)$ as well. Thirdly, I check for any $Z(\boldsymbol{x}) \leq p$ for $\boldsymbol{x} \in Q_n$ and for all $p$ in the given range and this action works in $O(n)$. And lastly, we repeat the above for $m$ different realizations which means the complexity of the program in total will be $O\big(m(n^2+n)\big) \approx O(mn^2)$. We can deduce that the computational time for the program is also in $O(mn^2)$.

Therefore, I have chosen a smaller value for $n$ compared with $m$ at the start of the question.

As it is more likely for $|C_n| = \emptyset$ when $n \to \infty$ for p fixed, we expect $p_c$ increases and getting closer to 0.644 as n increases and the graph of $\theta(\mathrm{p})$ shifts to the right compared with the graph we have above.




Graph 2: plotting $\hat{\theta}_{200,2000}$ $and$ $\hat{\theta}_{200,1000}$ $against\ p$

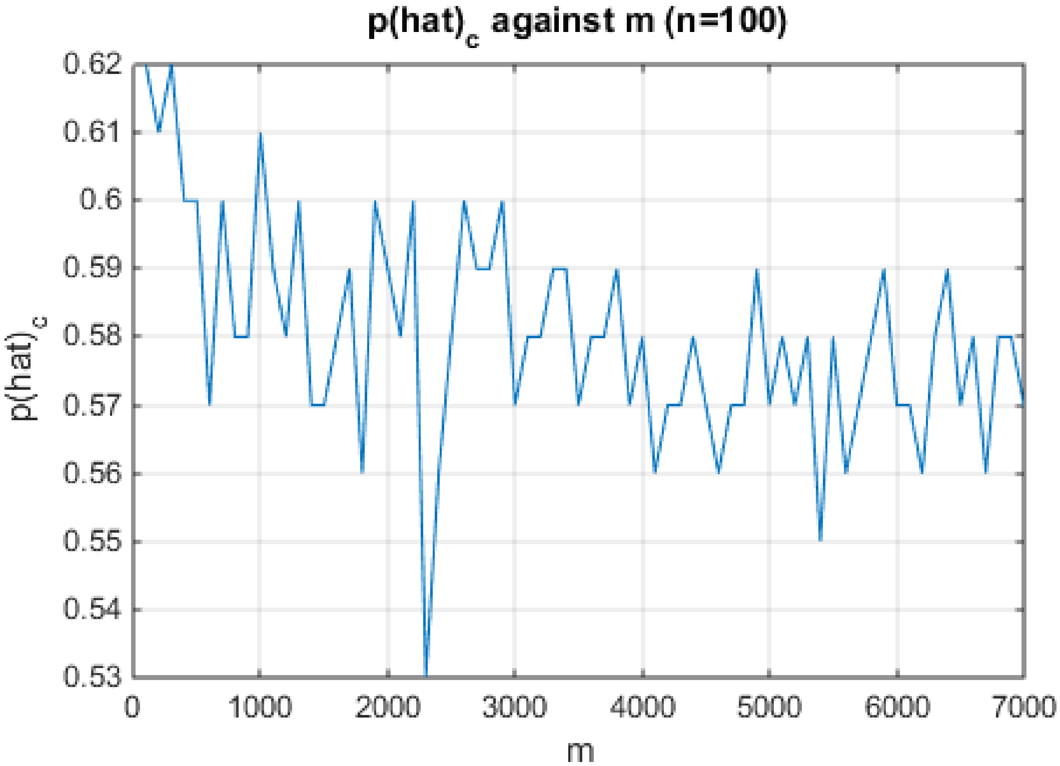
Comparing two graphs with $\hat{\theta}_{200,2000}$ $and$ $\hat{\theta}_{200,1000}$ in Graph 2, we can see $p_c$ shifts from 0.62 to 0.63, which is closer to the true value, 0.644 as n increases. In addition, we can see the graph behaves like the error function and tends to a Heaviside function as n increases. I believe as n increases, the graph will tend to $H(p - 0.644)$ where H is a Heaviside function.

## Question 3:

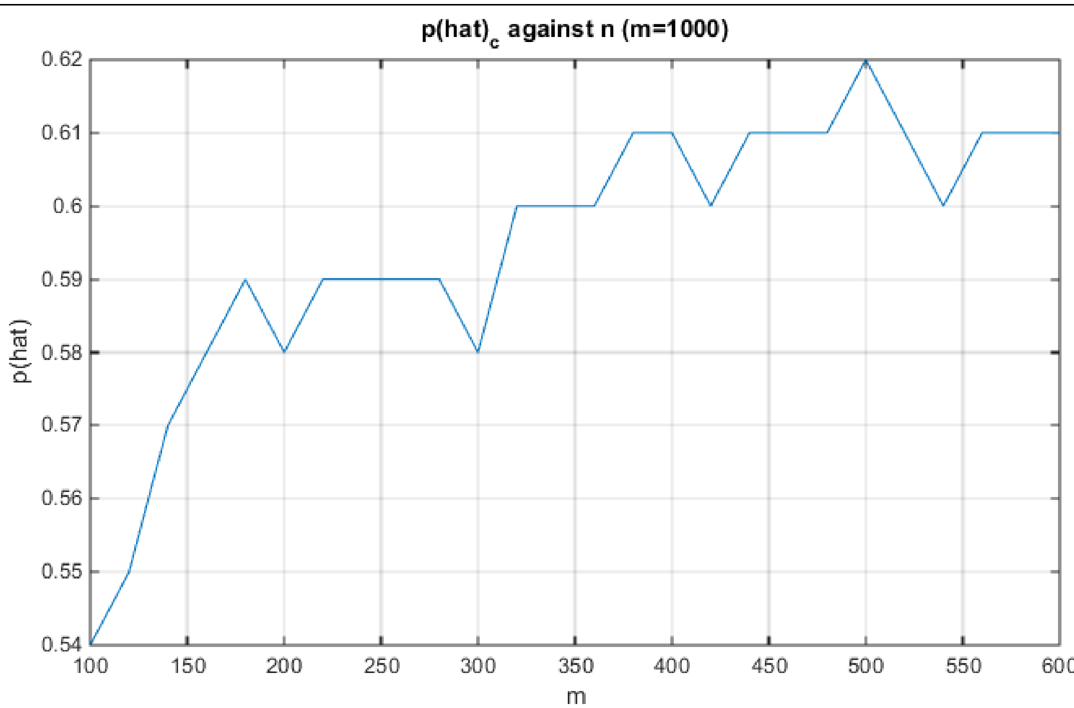
From graph 4, we can see $\hat{p}_c$ increases and tends to reach the true value $p_c = 0.644$ as $n$ increases and $m$ (large) fixed. This is because, similar reason in Question 2, it is more likely for $C_n = \emptyset \; as \; n \; increases$. Therefore, the value for $\hat{p}_c$ increases as n increases. On the other hand, as $\widehat{\theta}_{m,n}(p)$ is a decreasing function in n and increasing function in p, for getting the $\mathrm{p_s} = \sup\{p\colon \widehat{\theta}_{m,n}(p) = 0\}$ for fixed m and increasing n, we should expect $p_s$ increases and it will always be $p_s(n) \leq p_c$ and $p_s(n) \to p_c$ for $n \to \infty$.

For fixed n, as m increases, we expect $\hat{p}_c$ fluctuate and converges to a limit $\hat{p}_c(n)$ where $\hat{p}_c(n)$ is a constant and $\hat{p}_c(n) = \sup\{p\colon \theta_\mathrm{n}(p) = 0\} \leq p_c$. ('Less than or equal to $\mathrm{p_c}$' because of squared statement above.). For showing the convergence of $p_s = \sup\{p\colon \widehat{\theta}_{m,n}(p) = 0\} \to \hat{p}_c(n) = \sup\{p\colon \theta_\mathrm{n}(p) = 0\} \; as \; m \to \infty$, we may use the statement from the law of large numbers where the sample average tends to the expected value as $m \to \infty$. We can confirm our prediction with the Graph 3 as we can see a fluctuation between 0.5 to 0.57 and converging to the range of [0.51, 0.54] where $p_s$ is.

From both graph, as we achieve our approximation by averaging m realizations with 2 pseudo-random matrices, it is normal to see fluctuation, however, we can still see the tendency of the graph.




Graph 3: plotting $p_c$ $against\ m\ with\ n = 100$




Graph 4: plotting $p_c$ $against\ n\ with\ m = 1000$

## Question 4:

To estimate $\gamma$, I have written a program [Q4.m] which gives me the $P_p(C_n \neq \emptyset)$ out of the m realization with a given p through running the program [percolationQ2_mod.m] and calculate $\gamma$ with the calculated $P_p(C_n \neq \emptyset)$ and input n.

It is tricky to pick the suitable value for m and n as it always turns out to be undefined if I have picked a relatively large n due to $P_p(C_n \neq \emptyset) = 0$ if n is large and p is in the range [0.3-0.6] and $\log(0)$ does not give us any useful information at all. Therefore, I need to pick a relative small n for this estimation.

On the other hand, as there is more chance for me to get at least a realization with $(C_n \neq \emptyset)$ if I picked a large m, I expect to pick a large m in respect to n for the estimation.

As $P_p(C_n \neq \emptyset)$ an increasing function in p and decreasing function in n, I expect that we can increase our value of n as we increase in p. By trial and error, I obtained the largest n which works for the smallest p for a large m=100000. To avoid unreasonably long computation time, as the program in $O(mn^2)$, I have made adjustment of the value of m according to the value of n.

| $p$ | $\gamma$ | m | n |
|---|---|---|---|
| 0.3 | 0.5207 | 100000 | 20 |
| 0.4 | 0.3005 | 50000 | 36 |
| 0.5 | 0.1012 | 30000 | 95 |
| 0.6 | 0.0186 | 10000 | 250 |

In theory, I expect to see $\gamma$ decreases as p is reaching $p_c$ from below. This is because $P_p(C_n \neq \emptyset) \rightarrow 1$ if p is higher so we expect $\gamma \rightarrow 0$ as p increases.

## Reference:

1. http://www.stat.ucla.edu/~nchristo/introeconometrics/introecon_central_limit_theorem.pdf
   'The Central Limit Theorem'

## **Program:**

1:

```
function [W]=percolationQ2_mod(m,n,pstart,pend)
%m is the number of repeats; n is value of how far we
%approximate)
%p start and p end indicates range of p for the graph and
approximation
tic;
W=[];
for y=1:m,
%generate different matrix first
map=rand(n,n,2);
T=nan(n,n);
T(1,1)=0;
%1 is horizontal 2 is vertical
%coord in T is i+1,j+1
%define boundary condition
for t=2:n,
    T(1,t)=max(T(1,t-1),map(1,t-1,1));
    T(t,1)=max(T(t-1,1),map(t-1,1,2));
end
%calculate the probability of each node being accessible
for k=2:n,
    for u=2:(n-k+1)
        ver=max(T(k-1,u),map(k-1,u,2));
        hor=max(T(k,u-1),map(k,u-1,1));
        T(k,u)=min(ver,hor);
    end
end
O=0;
l=0;
e=[];
%check any nodes at distance n being accessible for different p
for p=pstart:0.01:pend,
    l=l+1;
    e=[e;p];
for z=1:n,
    if T(z,n-z+1)<=p
        O=1;
    end
end
W(l,y)=O;
end
end
W=sum(W,2)./m;
```

```
toc;
plot(e,W)
end
```

2:

```
function Q3fixm_mod(m)
T=[];
for n=100:20:600,
W=percolationQ2_mod(m,n,0.5,0.7);
W=sum(W,2);
K=find(W);
P=0.5+(K(1)-1)*0.01;
T=[T;n,P];
end
plot (T(:,1),T(:,2))
```

3:

```
function Q3fixn_mod(n)
T=[];
for m=100:100:7000,
W=percolationQ2_mod(m,n,0.45,0.75);
W=sum(W,2);
K=find(W);
P=0.5+(K(1)-1)*0.01;
T=[T;m,P];
end
plot (T(:,1),T(:,2))
```

4:

```
function [K]=Q4(m,n,p)

[W]=percolationQ2_mod(m,n,p,p);

K=(log(W(1)))/n;
end
```