Faculty Development Programme on

Network Science: Foundation Of Social Network Analysis

Generative Models

Hands-on Session (Day 4)

Swagata Duari
Kirti Jain

Real World Network : FACEBOOK

- Download Facebook graph : https://snap.stanford.edu/data/ego-Facebook.html

Reading edge list data from file :

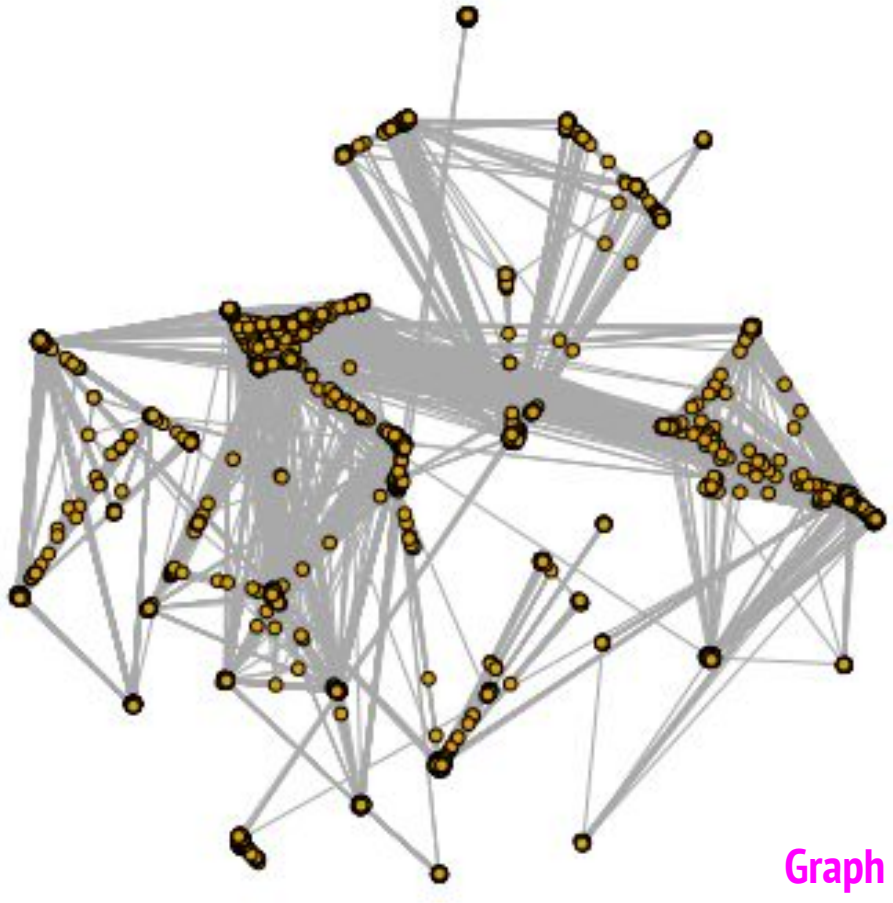
```
> el = read.csv ( "facebook_combined.txt",  sep = "", head=F)
```

Turning data frame(having edge list) into graph object :

```
> network <- graph_from_data_frame ( el , dir = FALSE)
> network
```

```
IGRAPH UN-- 4039  88234 --
+ attr: name (v/c)
+ edges (vertex names):
 [1] 0--1  0--2  0--3  0--4  0--5  0--6  0--7  0--8  0--9  0--10
0--11 0--12 0--13 0--14 0--15 0--16 0--17 0--18 0--19 0--20
0--21 0--22 0--23 0--24 0--25 0--26 0--27 0--28 0--29 0--30
0--31 0--32 0--33 0--34 0--35 0--36 0--37 0--38 0--39 0--40
0--41 0--42 0--43 0--44 0--45 0--46 0--47 0--48 0--49 0--50
0--51 0--52 0--53 0--54 0--55 0--56 0--57 0--58 0--59 0--60
0--61 0--62 0--63 0--64 0--65 0--66 0--67 0--68 0--69 0--70
0--71 0--72 0--73 0--74 0--75 0--76 0--77 0--78 0--79 0--80
0--81 0--82 0--83 0--84 0--85 0--86 0--87 0--88 0--89 0--90
0--91 0--92 0--93 0--94 0--95 0--96 0--97 0--98 0--99
+ ... omitted several edges
```

> ***plot*** (network)



Number of Nodes

> **vcount** (network)

OUTPUT : 4039

Number of Edges

> **ecount** (network)

OUTPUT : 88234

Diameter

> **diameter** (network)

OUTPUT : 8

Average Shortest Path Length 'μ'

> **mean_distance** (network)

OUTPUT : 3.692

Log n

> **log** (**vcount** (network))

OUTPUT : 8.303

Graph exhibits *ultra-small world* behaviour $\mu <<< \log n$.

Demonstration of Scale-Free Property of Graph

Get the degree of the graph:

```
> d <- degree ( network , mode="all")
```

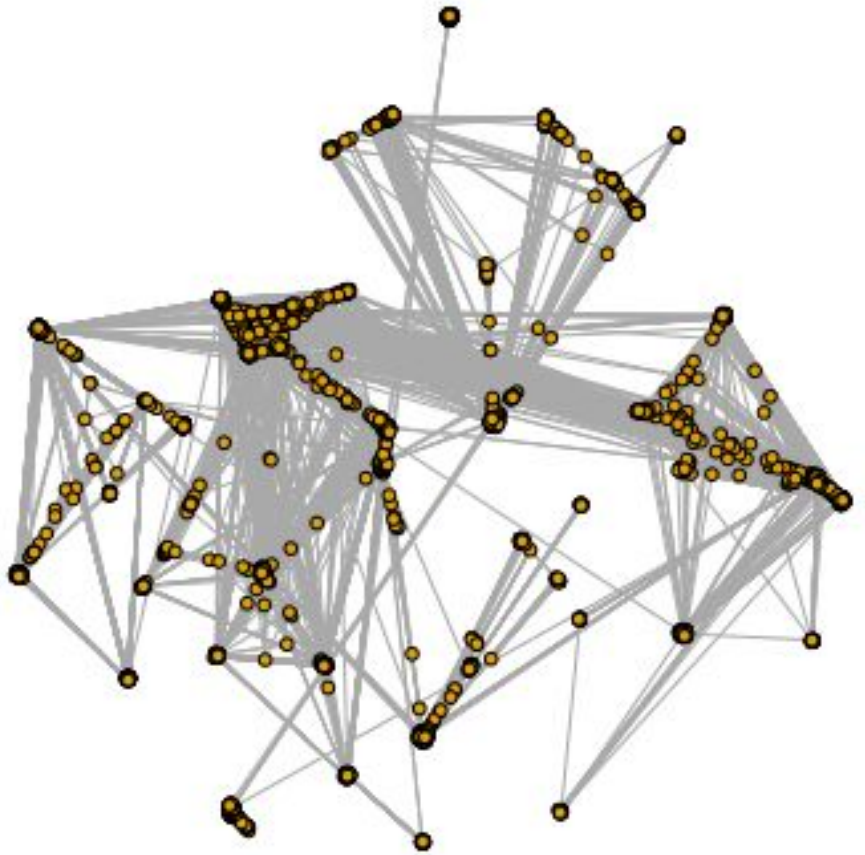
Fits power-law distribution to degree:

```
> power <- power.law.fit ( d )
```

The power-law exponent:

```
> power $ alpha
```

OUTPUT : **2.510175**

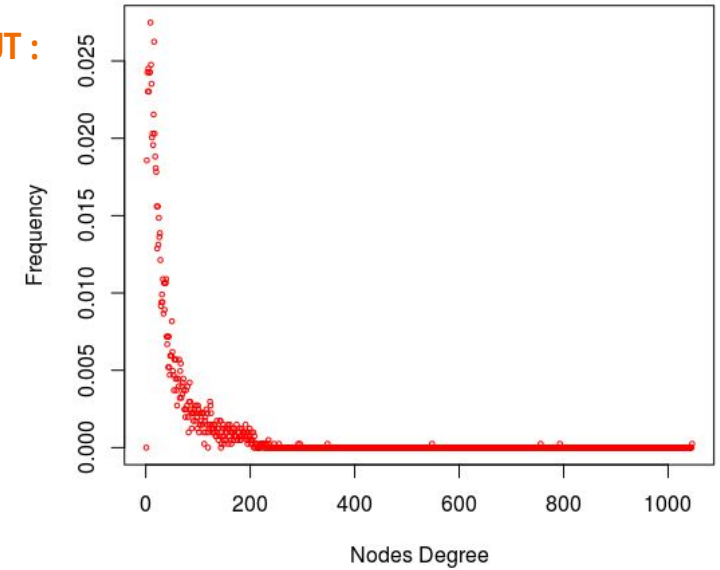


Degree Distribution of Facebook Network

```
> dd = degree.distribution ( g, mode = "all", cumulative = FALSE)
```

```
> plot(dd, ylim=c(0,max(dd)),xlab="Nodes Degree", ylab="Frequency")
```

OUTPUT :

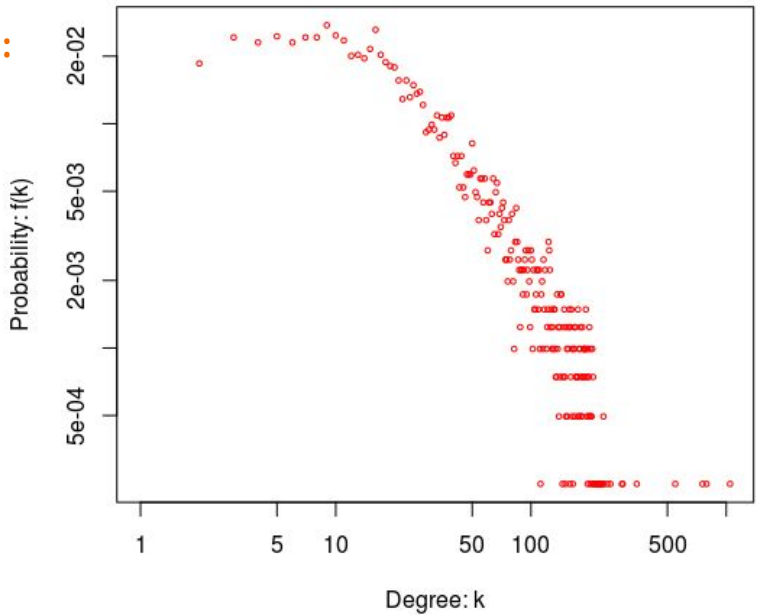


Plot shows that vast majority of nodes have very small degrees, whereas there are a few "hub" nodes that have high degrees.

Log-log plot of degree (k) versus degree distribution f(k) :

```
> plot ( dd,log='xy', xlab="Degree: k", ylab="Probability: f(k)")
```

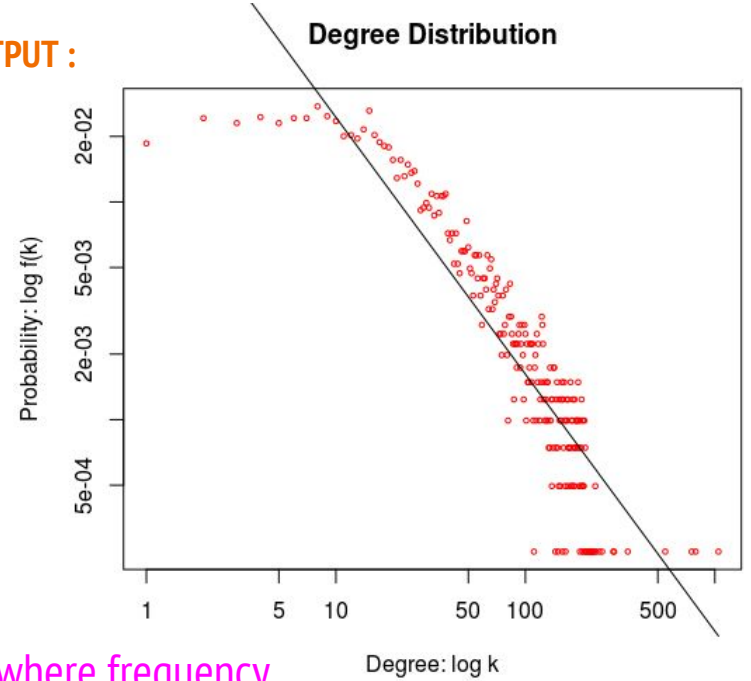
OUTPUT :



Use the newly defined function to plot the log-log plot :

```
> fit_power_law ( dd, d, 'Degree Distribution')
```

OUTPUT :



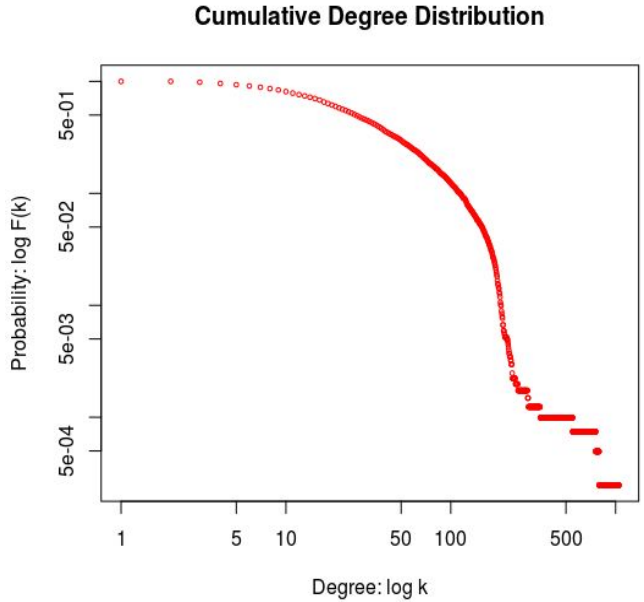
Note that there is **high level of noise for the higher degrees**, where frequency counts are the lowest. Solution: use the **Cumulative degree distribution F(k).**

Cumulative degree distribution F(k) :

```
> cdd = degree.distribution(g, mode = "all", cumulative = TRUE)
```

```
> plot( cdd,log='xy', xlab="Degree: log k", ylab="Probability: logF(k)",
      main="Cumulative Degree Distribution")
```

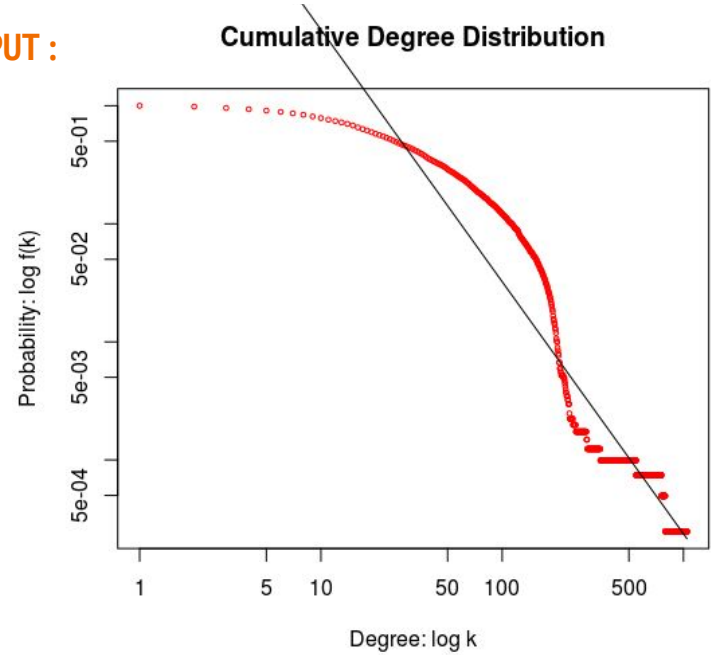
OUTPUT :



Use the newly defined function to plot the log-log plot :

```
> fit_power_law ( cdd, d, 'Cumulative Degree Distribution')
```

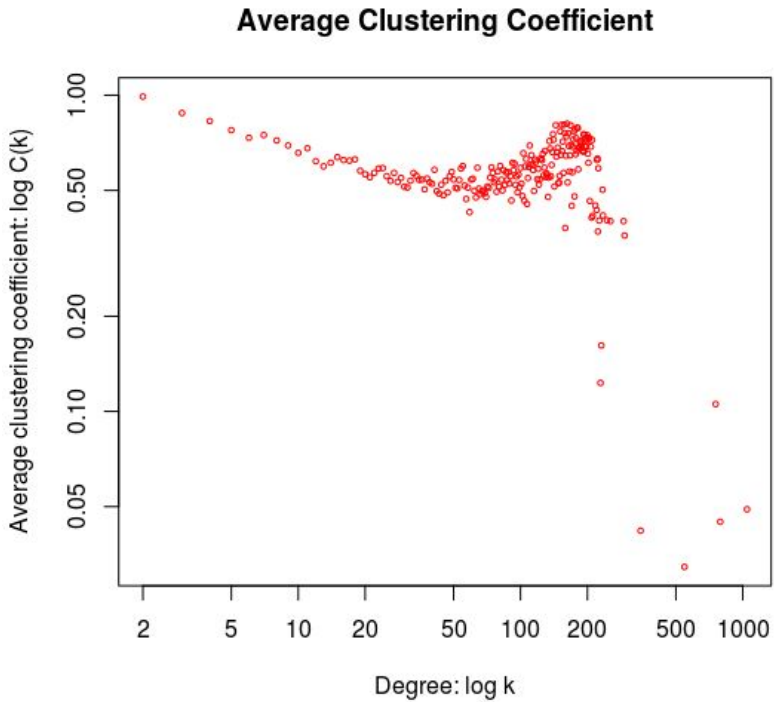
OUTPUT :



Clustering Coefficient of Facebook Network

> **transitivity** (network)

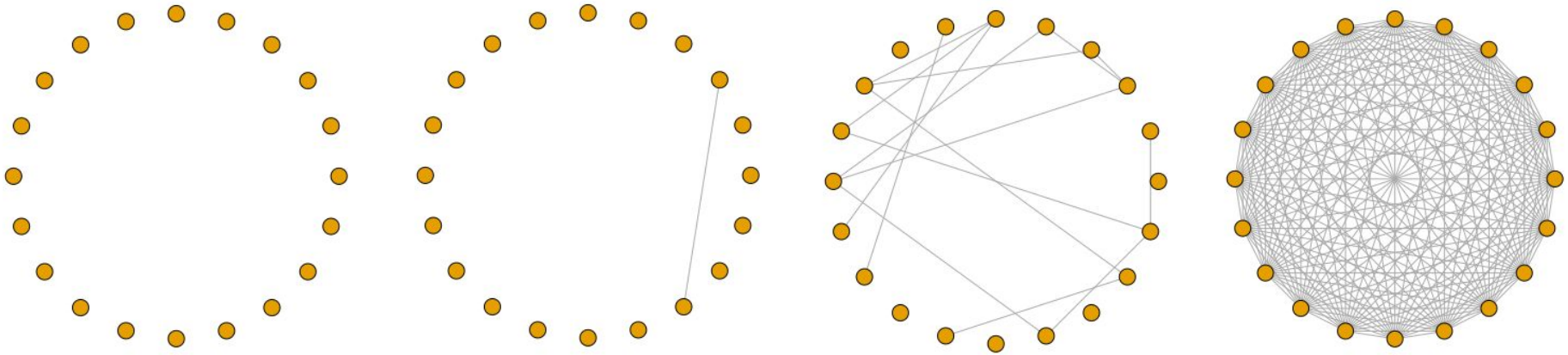
OUTPUT : 0.5191743



Erdos-Renyi (ER) Model

Generate random graphs according to the Erdos-Renyi model.

```
> erg <- erdos.renyi.game ( 20 , 0 ,  directed = FALSE )  #no edges
> plot ( erg , layout = layout_in_circle, vertex.size=10 )
```



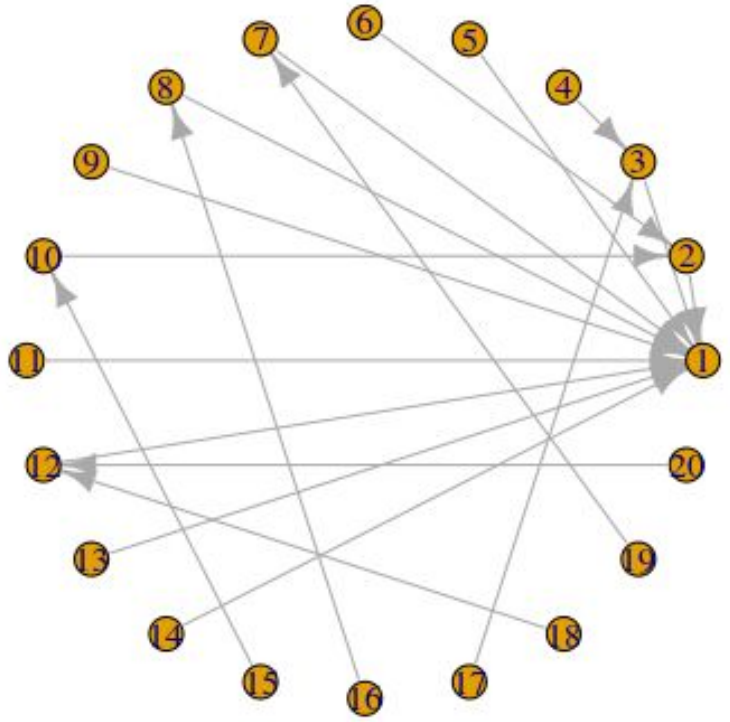
P = 0 (empty) P = 0.01 P = 0.05 P = 1

Increase the probability of connecting edges.

Barabasi-Albert preferential attachment Model (BA) Model

Generate scale-free graphs.

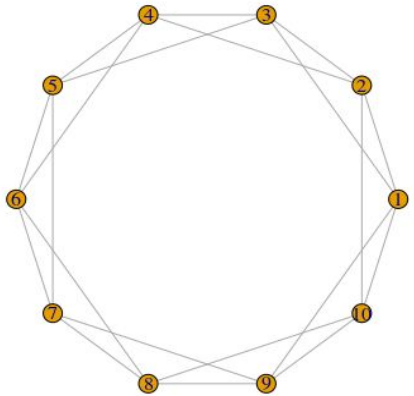
```
> bag <- sample_pa ( 20 )
> plot ( bag , layout = layout_in_circle, vertex.size=9 )
```



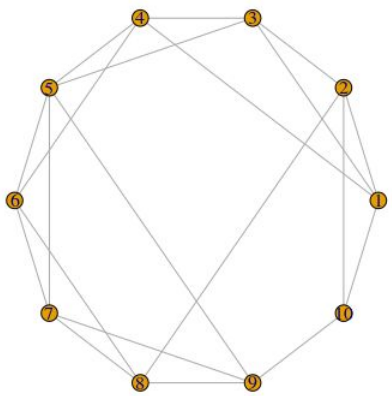
Watts-Strogatz (WS) Model

Generate a graph according to the Watts-Strogatz network model.

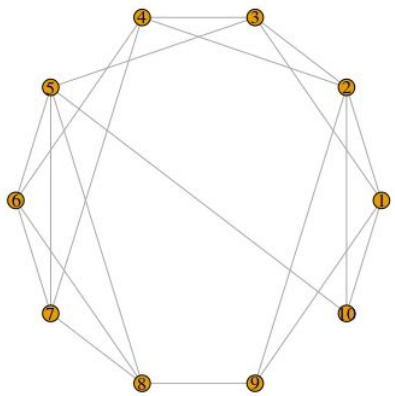
```
> wsg <- watts.strogatz.game ( dim=1, size=10, nei=2, p=0)
> plot ( wsg , layout = layout_in_circle )
```



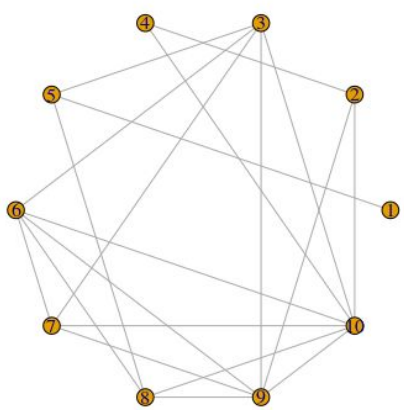
P = 0 (Regular)



P = 0.05



P = 0.1



P = 1 (Random)

Increase the probability of rewiring edges. (more randomness)

Thankyou !