

HomeWork 4

{Your Name}

deadline - 01.06.2015

Contents

Power law. Descriptive network analysis	1
Problem 1	1
Problem 2	2
Problem 3.	3

Power law. Descriptive network analysis

Use this file as a template for your report.
Support your computations with figures and comments.
Send ONLY hw4_Surname.Rmd versions of your report.

```
#install.packages('igraph')  
library(igraph)
```

Problem 1

Recall from the lecture that probability density function (PDF) for power law distributed variable is:

$$p(x) = Cx^{-\alpha}$$

Take logarithm of both sides:

$$\log p(x) = \log C - \alpha \log x$$

Now you can use standard R functions like `lm()` to calculate α coefficient via linear regression. However you might find that it is a bad idea.

Alternatively, you can compute cumulative density function (CDF)

$$f(x) = Pr(x < X)$$

of power law distribution. Good things about CDF of the power law are:

- It still has form of power law
- On log-log plot it looks more like a line

1. Derive the formula for CDF function of power law $F(x)$. It means you should calculate cumulative distribution function by integration of PDF function. *write your answer here*

```
## Put your code here
```

2. Download [Internet Network](#) and plot PDF and CDF of the degree distribution in log-log scale

```
## Put your code here
```

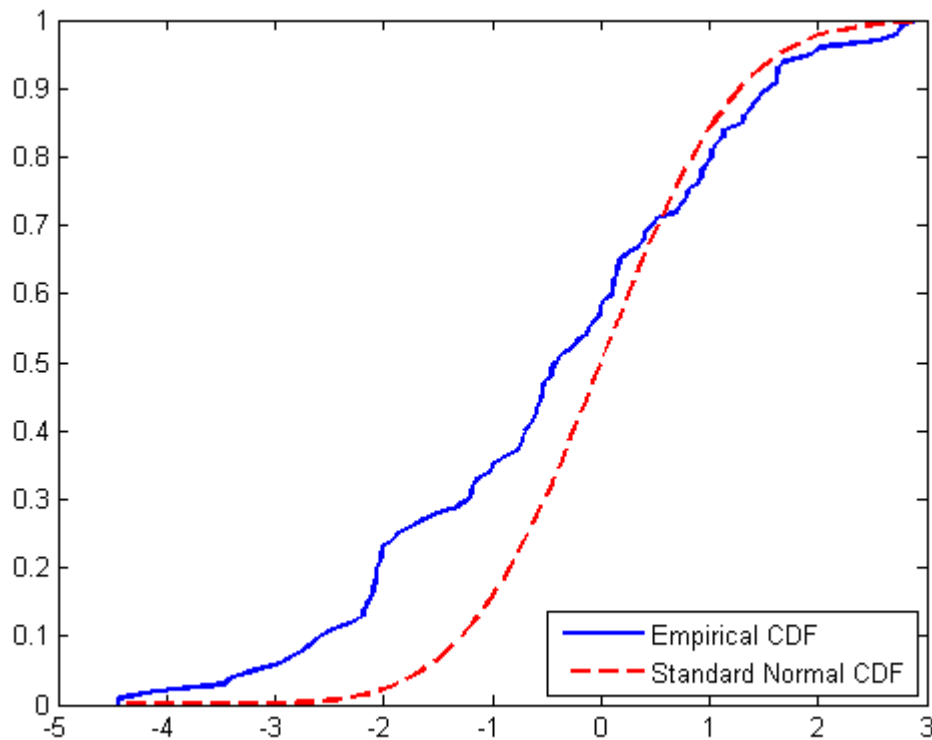
3. Fit linear regression model to PDF and CDF to estimate α . Plot fitted models along with data

Problem 2

Kolmogorov-Smirnov test describes how similar are two distributions. In our case, when we have fitted model and original data, we can calculate their CDFs and Kolmogorov-Smirnov test shows us how well model approximates original data. In other words, it shows us the goodness-of-fit of our model.

$$D = \max_x \|f(x|\alpha, x_{min}) - f_{emp}(x)\|,$$

where $f(x|\alpha, x_{min})$ and $f_{emp}(x)$ are theoretical and empirical CDFs respectively.



To estimate x_{min} of the fitted power-law model we can use KS test:

- Pick some x_{min} value
- Fit power-law distribution to data (that is estimation of α) – now we have $f(x|\alpha, x_{min})$
- Perform KS test – compute D statistic
- Finally, choose x_{min}^* that provides minimal value of D statistic among all KS tests run above.

In R all this stuff can be done in one line of code.

Again, use [Internet Network](#)

Properly load it into R and do following tasks:

```
## Put your code here
```

1. Using `power.law.fit` find `xmin` value and corresponding `alpha`

```
## Put your code here
```

2. Put fitted model along with empirical PDF (CDF)

Problem 3.

For [Wikipedia vote network](#) (clear up comments in the begging of the file) derive the following characteristics:

1. The number of vertices and edges
2. The number of loops (edges that start and end at the same vertex)
3. The number of symmetrical edges
4. Degree distribution (without considering the direction)
5. The number of nodes with a degree greater than 1 and with a degree greater than 15
6. Find strongly connected components and their sizes.
7. Take subgraph of the original graph, which consists of the first 80 vertices and set color into red for those nodes in which the number of incoming edges is greater than the number of outgoing edges. Otherwise, set color in blue. For nodes with the same number of incoming and outgoing edges set color into green. Besides that, increase the size of vertices with a maximum value of transitivity (for example, you may set size into 10 for these nodes and 1 for others).
8. Take subgraph from the previous task and find maximal connected component. For this component highlight any way that corresponds to the diameter of the subgraph. How many such paths are in this graph?
9. Make average neighbor degree vs node degree scatter plot (one point on the plot per node) and aggregated plot, averaging over all nodes with the same degree (aggregated average vs degree, one value per degree). Explain your observations.
10. Make local clustering coefficient vs node degree scatter plot (one point on the plot per node) and aggregated, averaging over all nodes with the same degree (aggregated average vs degree, one value per degree). Explain your observations.

```
## Put your code here
```

1. The number of vertices and edges.

```
## Put your code here
```

2. The number of loops (edges that start and end at the same vertex)

```
## Put your code here
```

3. The number of symmetrical edges

```
## Put your code here
```

4. Degree distribution

```
## Put your code here
```

5. The number of nodes with a degree greater than 1 and with a degree greater than 15

```
## Put your code here
```

6. Find strongly connected components and their sizes.

```
## Put your code here
```

7. Take subgraph of the original graph, which consists of the first 80 vertices and set color into red for those nodes in which the number of incoming edges is greater than the number of outgoing edges. Otherwise, set color in blue. For nodes with the same number of incoming and outgoing edges set color into green. Besides that, increase the size of vertices with a maximum value of transitivity (for example, you may set size into 10 for these nodes and 1 for others).

```
## Put your code here
```

8. Take subgraph from the previous task and find maximal connected component. For this component highlight any way that corresponds to the diameter of the subgraph. How many such paths are in this graph?

```
## Put your code here
```

9. Make average neighbour degree vs node degree scatter plot (one point on the plot per node) and aggregated plot, averaging over all nodes with the same degree (aggregated average vs degree, one value per degree). Explain your observations.

```
## Put your code here
```

10. Make local clustering coeff vs node degree scatter plot (one point on the plot per node) and aggregated, averaging over all nodes with the same degree (aggregated average vs degree, one value per degree). Explain your observations.