

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)**

Факультет Инфокоммуникационных технологий (ИКТ)

Образовательная программа Мобильные и сетевые технологии

О Т Ч Е Т
по Лабораторной работе

Дисциплина: Методы сетевого анализа.

Специальность: 09.03.03 Прикладная информатика.

Выполнил:

Балцат К. И.,

студент группы К33401

Санкт-Петербург
2023

ЗАДАНИЕ

Для того, чтобы строить прогнозы о дальнейшем поведении сообщества, будет полезно определить, по принципу какой модели строятся отношения между его участниками. Для выполнения работы вам понадобятся социальные графы, отражающие взаимоотношения участников 2 сообществ социальной сети В контакте (можно использовать сформированные в ходе выполнения первой лабораторной работы).

1. Постройте функции распределения степеней узлов для обоих сообществ

На какую из известных вам функций похожа каждая из них? В отчёте отразите формулу, график.

2. Вычислите кластерные коэффициенты обоих сообществ

Кластерный коэффициент (локальная плотность) - вероятность того, что два соседа в графе связаны между собой

3. Вычислите среднюю длину пути обоих графов

4. Изучите существующие модели социальных сетей. Каковы характерные для них диапазоны оценок? Какую (или какие) из них напоминают исследуемые графы?

На практике не всегда получается подогнать сообщество под одну из существующих моделей, но вычисленные вами метрики помогут оценить сходство. Оцените, попадают ли получившиеся значения метрик в диапазоны оценок известных моделей, и с какой точностью определены характерные значения в вашем случае. В этом вам может помочь следующая таблица:

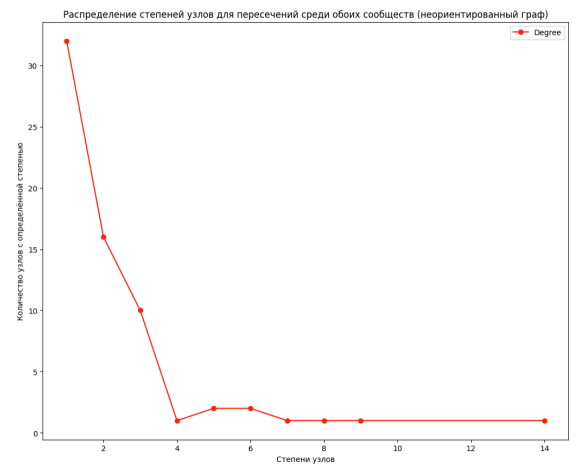
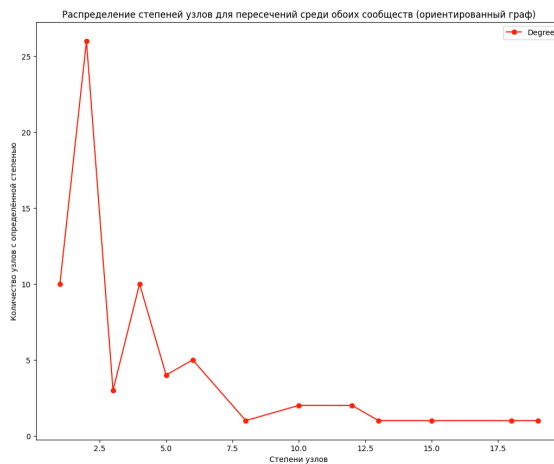
	Random	BA model	WS model	Empirical networks
$P(k)$	$\frac{\lambda^k e^{-\lambda}}{k!}$	k^{-3}	poisson like	power law
C	$\langle k \rangle / N$	$N^{-0.75}$	const	large
$\langle L \rangle$	$\frac{\log(N)}{\log(\langle k \rangle)}$	$\frac{\log(N)}{\log \log(N)}$	$\log(N)$	small

ВЫПОЛНЕНИЕ

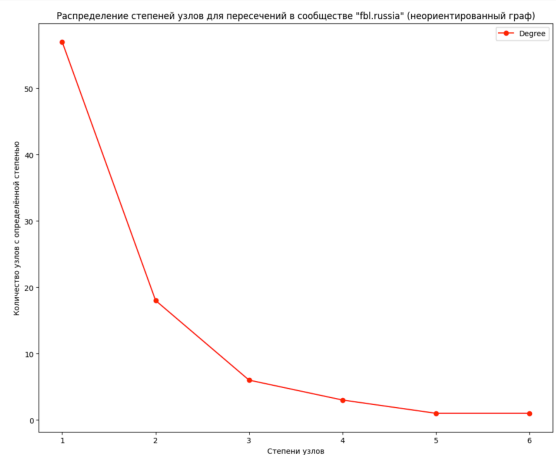
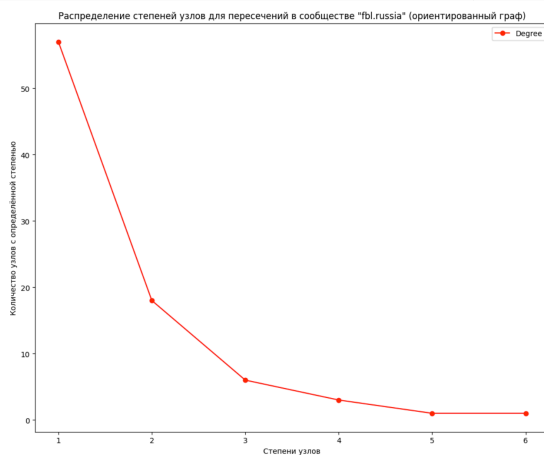
Задание №1. Функции распределения степеней узлов

Задание №1. Функции распределения степеней узлов

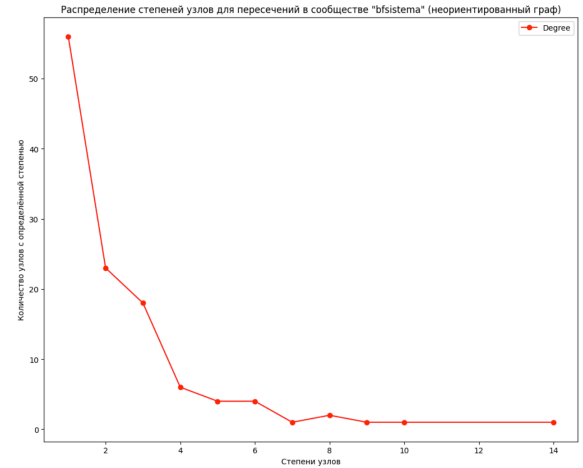
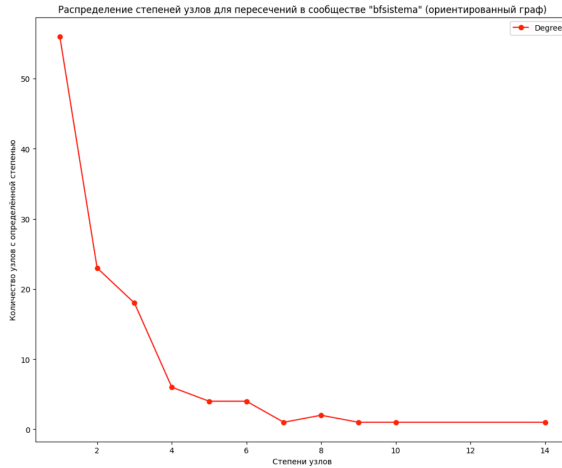
```
[27] degree = dict(G.degree())
degree_values = sorted(list(degree.values()))
hist = [list(degree_values().count(x) for x in degree_values)]
degree1 = dict(G1.degree())
degree_values1 = sorted(list(degree1.values()))
hist1 = [list(degree1.values().count(x) for x in degree_values1)]
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(27.5, 10))
ax1.plot(degree_values, hist, 'ro-')
ax1.legend(['Degree'])
ax1.set_xlabel('Степени узлов')
ax1.set_ylabel('Количество узлов с определенной степенью')
ax1.set_title('Распределение степеней узлов для пересечений среди обоих сообществ (ориентированный граф)')
ax2.plot(degree_values1, hist1, 'ro-')
ax2.legend(['Degree'])
ax2.set_xlabel('Степени узлов')
ax2.set_ylabel('Количество узлов с определенной степенью')
ax2.set_title('Распределение степеней узлов для пересечений среди обоих сообществ (неориентированный граф)')
plt.show()
```



```
[30] degree = dict(G2.degree())
degree_values = sorted(list(degree.values()))
hist = [list(degree_values().count(x) for x in degree_values)]
degree1 = dict(G2_1.degree())
degree_values1 = sorted(list(degree1.values()))
hist1 = [list(degree1.values().count(x) for x in degree_values1)]
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(27.5, 10))
ax1.plot(degree_values, hist, 'ro-')
ax1.legend(['Degree'])
ax1.set_xlabel('Степени узлов')
ax1.set_ylabel('Количество узлов с определенной степенью')
ax1.set_title('Распределение степеней узлов для пересечений в сообществе "fbl.russia" (ориентированный граф)')
ax2.plot(degree_values1, hist1, 'ro-')
ax2.legend(['Degree'])
ax2.set_xlabel('Степени узлов')
ax2.set_ylabel('Количество узлов с определенной степенью')
ax2.set_title('Распределение степеней узлов для пересечений в сообществе "fbl.russia" (неориентированный граф)')
plt.show()
```



```
[31] degree = dict(G3.degree())
degree_values = sorted(set(degree.values()))
hist = [list(degree.values()).count(x) for x in degree_values]
degree1 = dict(G3_1.degree())
degree_values1 = sorted(set(degree1.values()))
hist1 = [list(degree1.values()).count(x) for x in degree_values1]
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(27.5, 10))
ax1.plot(degree_values, hist, 'ro-')
ax1.legend(['Degree'])
ax1.set_xlabel('Степени узлов')
ax1.set_ylabel('Количество узлов с определённой степенью')
ax1.set_title('Распределение степеней узлов для пересечений в сообществе "bfsistema" (ориентированный граф)')
ax2.plot(degree_values1, hist1, 'ro-')
ax2.legend(['Degree'])
ax2.set_xlabel('Степени узлов')
ax2.set_ylabel('Количество узлов с определённой степенью')
ax2.set_title('Распределение степеней узлов для пересечений в сообществе "bfsistema" (неориентированный граф)')
plt.show()
```



Задание №2. Кластерные коэффициенты

▼ Задание №2. Кластерные коэффициенты

```
✓ [19] from networkx.algorithms import approximation
16s both = approximation.average_clustering(G1, trials=1000000, seed=1000)
fbl_russia = approximation.average_clustering(G2_1, trials=1000000, seed=1000)
bfsistema = approximation.average_clustering(G3_1, trials=1000000, seed=1000)
print(f"Кластерный коэффициент для пересечения обоих сообществ равен {both}")
print(f"Кластерный коэффициент для сообщества 'fbl.russia' равен {fbl_russia}")
print(f"Кластерный коэффициент для сообщества 'bfsistema' равен {bfsistema}")
```

Кластерный коэффициент для пересечения обоих сообществ равен 0.114902

Кластерный коэффициент для сообщества 'fbl.russia' равен 0.0

Кластерный коэффициент для сообщества 'bfsistema' равен 0.024341

Задание №3. Средняя длина пути

Так как графы не являются сильно связными, придётся вычислять среднюю длину пути для каждой отдельной вершины, после чего считать среднее расстояние.

```
✓ [37] both_list = []
0s fbl_list = []
bfsistema_list = []
for A in (G1.subgraph(a).copy() for a in nx.connected_components(G1)):
    both_list.append(nx.average_shortest_path_length(A))
for B in (G2_1.subgraph(b).copy() for b in nx.connected_components(G2_1)):
    fbl_list.append(nx.average_shortest_path_length(B))
for C in (G3_1.subgraph(c).copy() for c in nx.connected_components(G3_1)):
    bfsistema_list.append(nx.average_shortest_path_length(C))
print(f"Средняя длина пути для пересечения обоих сообществ равна {mean(both_list)}")
print(f"Средняя длина пути для сообщества 'fbl.russia' равна {mean(fbl_list)}")
print(f"Средняя длина пути для сообщества 'bfsistema' равна {mean(bfsistema_list)}")
```

Средняя длина пути для пересечения обоих сообществ равна 1.5992986557568674

Средняя длина пути для сообщества 'fbl.russia' равна 1.4646730462519937

Средняя длина пути для сообщества 'bfsistema' равна 1.6993210637224945

Задание №4. Средняя длина пути

Социальные графы сообществ по метрике коэффициента кластеризации и $\langle L \rangle$ больше описываются случайной моделью.

ВЫВОД

Я выполнил лабораторную работу и на практике освоил методы анализа сетей.