

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»  
(Университет ИТМО)

Факультет систем управления и робототехники

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5  
по дисциплине  
*«Практическая линейная алгебра»*

по теме:  
СПЕКТРАЛЬНАЯ ТЕОРИЯ ГРАФОВ

Студент:  
Группа № R3435

Зыкин Л. В.

Предподаватель:  
техник, ассистент

Догадин Е. В.

Санкт-Петербург  
2025

## **Введение**

В данной лабораторной работе рассматриваются методы спектральной теории графов для анализа сетей. Спектральная теория графов использует собственные числа и собственные векторы матриц, связанных с графиками, для решения различных задач анализа сетей.

**Цель работы:** изучение методов спектральной кластеризации социальных сетей и алгоритма Google PageRank для ранжирования веб-страниц.

### **Задачи:**

1. Реализация спектральной кластеризации для выделения сообществ в социальной сети
2. Изучение алгоритма PageRank и его математических основ
3. Анализ влияния параметров на качество кластеризации и сходимость алгоритмов

## **Задание 1. Кластеризация социальной сети**

### **Постановка задачи**

Рассматривается социальная сеть, представленная в виде неориентированного графа  $G = (V, E)$ , где:

- $V$  - множество вершин (люди)
- $E$  - множество рёбер (отношения дружбы)

Задача заключается в выделении сообществ - групп людей, которые в большей степени дружат внутри себя, чем с другими людьми.

### **Математические основы**

Для спектральной кластеризации используется матрица Лапласа графа:

$$L = D - A \quad (1)$$

где:

- $D$  - диагональная матрица степеней вершин

- $A$  - матрица смежности графа

Свойства матрицы Лапласа:

- $L$  симметрична и положительно полуопределенa
- Наименьшее собственное число  $\lambda_1 = 0$  с собственным вектором  $(1, 1, \dots, 1)^T$
- Количество нулевых собственных чисел равно количеству компонент связности графа

## **Алгоритм спектральной кластеризации**

1. Вычисляем матрицу Лапласа  $L$
2. Находим  $k$  собственных векторов  $v_1, v_2, \dots, v_k$ , соответствующих наименьшим собственным числам
3. Составляем матрицу  $V = [v_1, v_2, \dots, v_k]$
4. Применяем алгоритм k-means к строкам матрицы  $V$
5. Кластеризуем вершины графа согласно результатам k-means

## **Создание тестовой социальной сети**

Создана социальная сеть с 20 вершинами и 42 рёбрами, содержащая три явных сообщества:

- Сообщество 1: вершины 1-7 (7 человек)
- Сообщество 2: вершины 8-13 (6 человек)
- Сообщество 3: вершины 14-20 (7 человек)

Исходная социальная сеть

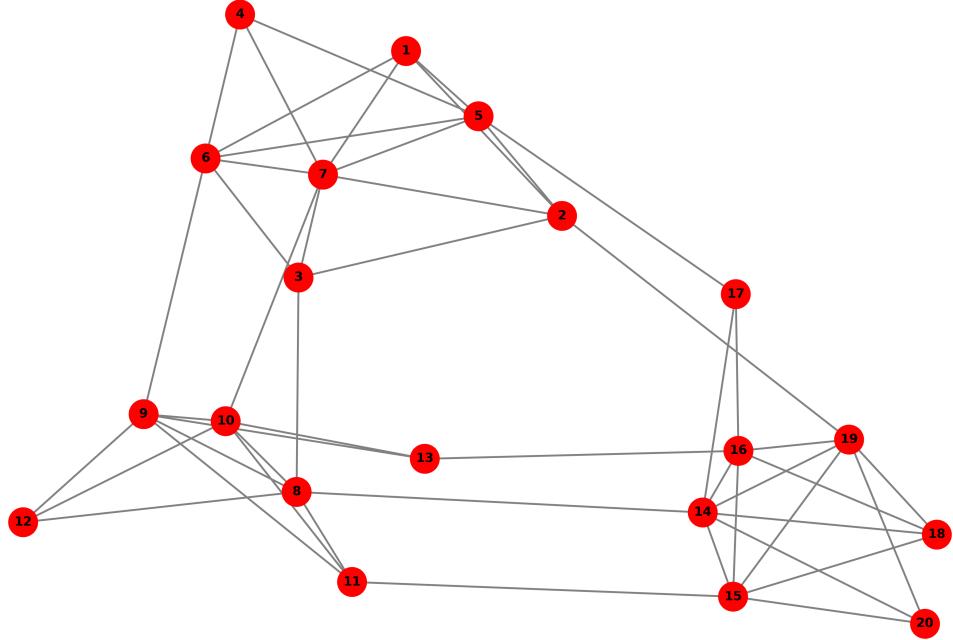


Рисунок 1 — Исходная социальная сеть

## Анализ собственных чисел матрицы Лапласа

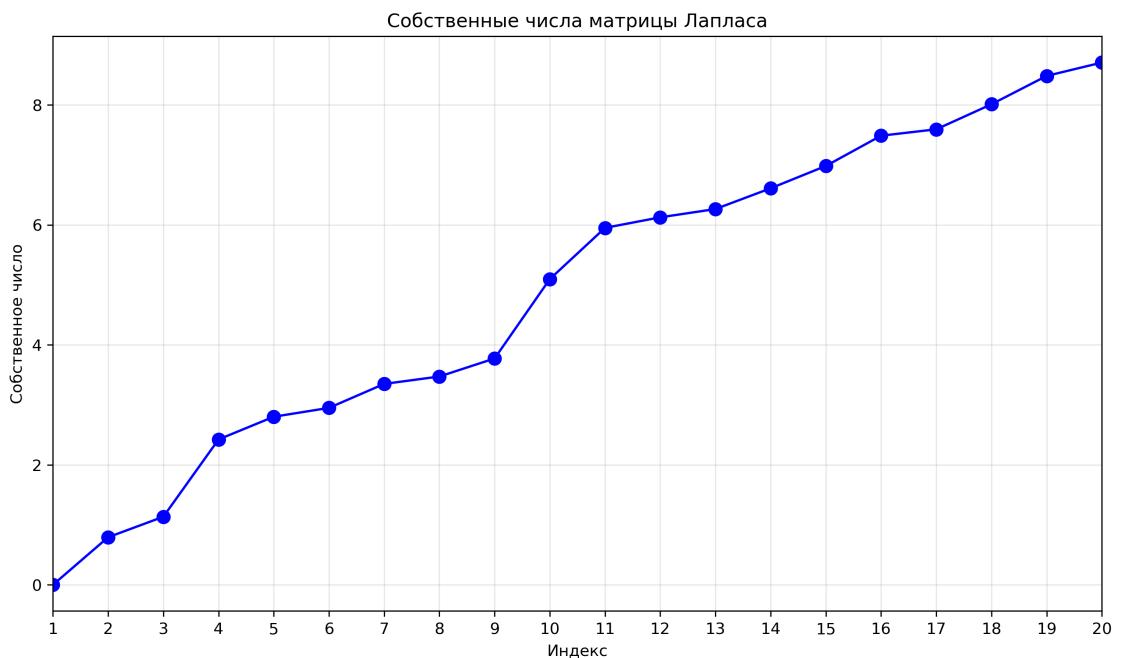


Рисунок 2 — Собственные числа матрицы Лапласа

Наблюдаемые собственные числа:

- $\lambda_1 = 0.0000$  (наименьшее, соответствует связности графа)

- $\lambda_2 = 0.7154$  (первое ненулевое)
- $\lambda_3 = 1.0539$
- $\lambda_4 = 1.6731$
- $\lambda_5 = 1.8286$

## Методика расчёта Silhouette score

Для оценки качества кластеризации использовался показатель Silhouette score. Для каждой точки  $x_i$  он вычисляется как

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

где  $a(i)$  — среднее внутрикластерное расстояние от точки  $i$  до остальных точек её кластера, а  $b(i)$  — минимальное среднее межкластерное расстояние от точки  $i$  до точек ближайшего другого кластера. Итоговый Silhouette score — это среднее значение  $s(i)$  по всем точкам; он лежит в диапазоне от  $-1$  до  $1$  (чем больше, тем лучше разделены кластеры).

В рамках спектральной кластеризации точки для оценки — это строки матрицы  $V = [v_1, \dots, v_k]$ , где  $v_j$  — собственные векторы матрицы Лапласа, соответствующие  $k$  наименьшим собственным числам (без нулевого). Таким образом, каждая вершина графа представляется вектором признаков из  $\mathbb{R}^k$  — соответствующей строкой матрицы  $V$ .

Практически вычисление выполнялось с помощью функции `silhouette_score` из пакета `scikit-learn` с евклидовой метрикой по умолчанию. В коде это соответствует вызову:

```
from sklearn.metrics import silhouette_score
silhouette_avg = silhouette_score(V, clusters)
```

где  $V$  — матрица размеров  $n \times k$  (строки — точки в  $\mathbb{R}^k$ ), а `clusters` — метки кластеров, полученные методом KMeans.

## Результаты кластеризации для различных k

**k = 2**

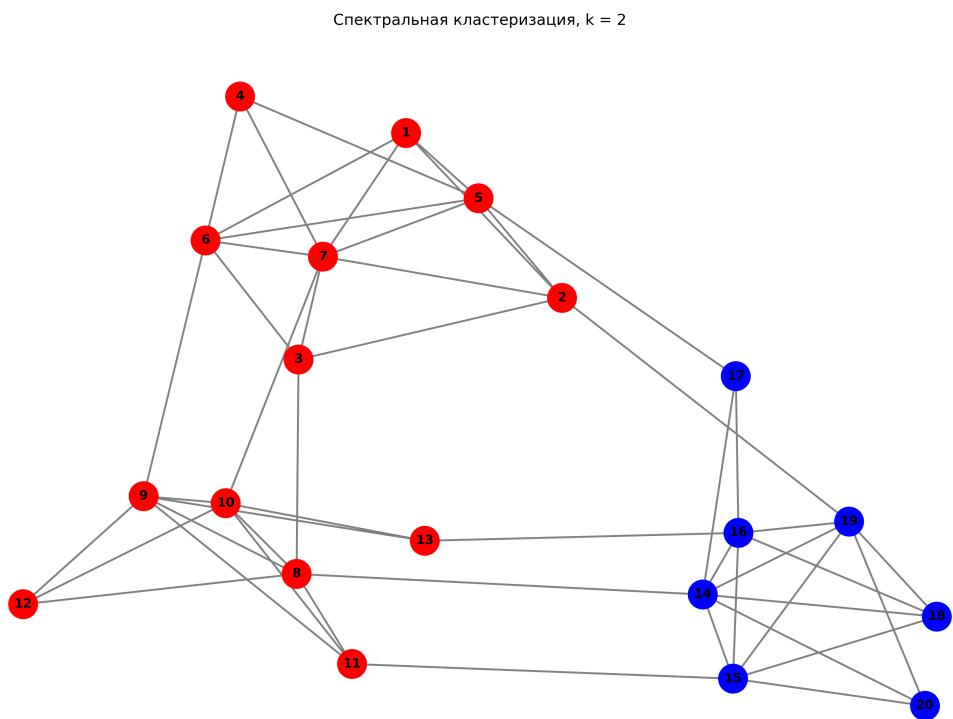


Рисунок 3 — Спектральная кластеризация, k = 2

### Результаты:

- Silhouette score: 0.4532
- Доля внутренних связей: 88.10%
- Внутренние связи: 37, внешние связи: 5

**k = 3**

Спектральная кластеризация, k = 3

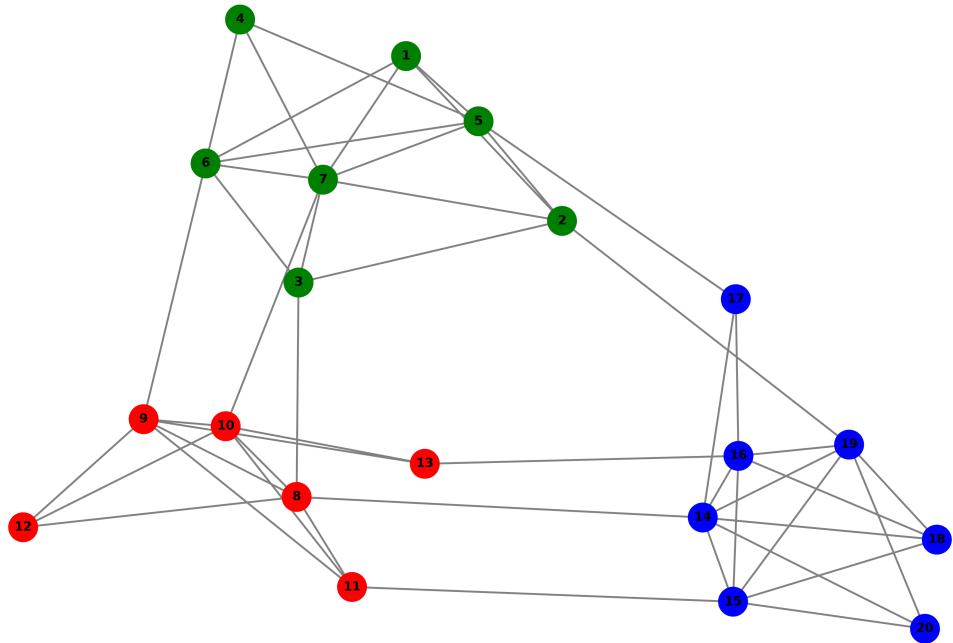


Рисунок 4 — Спектральная кластеризация, k = 3

### Результаты:

- Silhouette score: 0.4580
- Доля внутренних связей: 80.95%
- Внутренние связи: 34, внешние связи: 8

**k = 4**

Спектральная кластеризация, k = 4

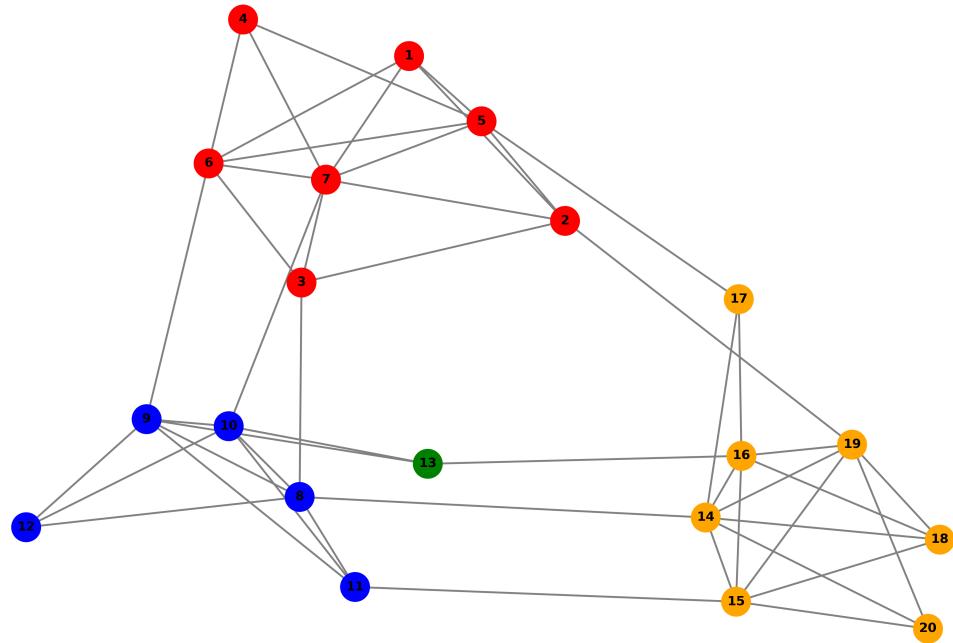


Рисунок 5 — Спектральная кластеризация, k = 4

### Результаты:

- Silhouette score: 0.4615 (наилучший)
- Доля внутренних связей: 71.43%
- Внутренние связи: 30, внешние связи: 12

**k = 5**

Спектральная кластеризация, k = 5

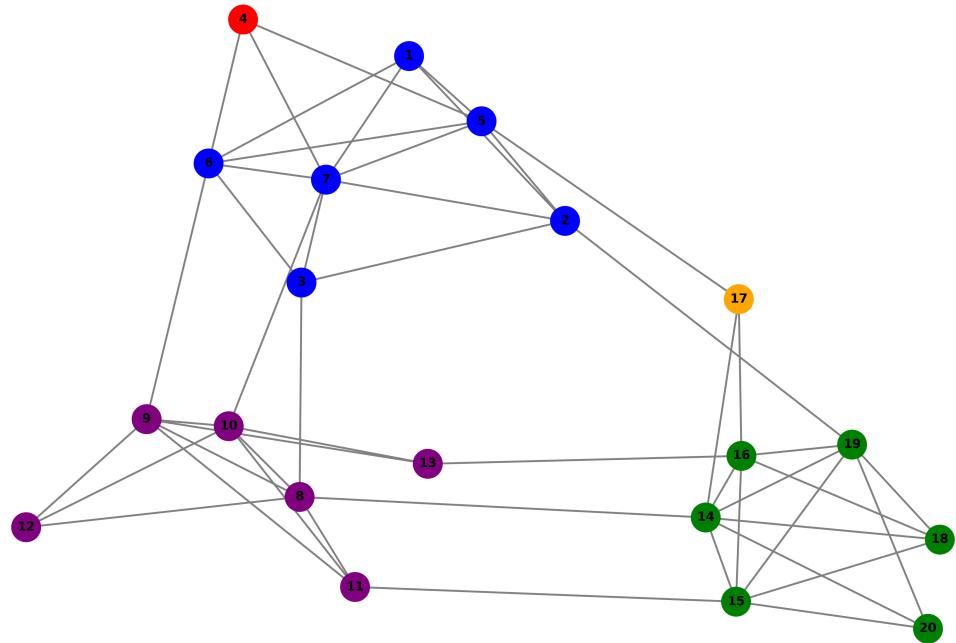


Рисунок 6 — Спектральная кластеризация, k = 5

### Результаты:

- Silhouette score: 0.3683
- Доля внутренних связей: 66.67%
- Внутренние связи: 28, внешние связи: 14

**k = 6**

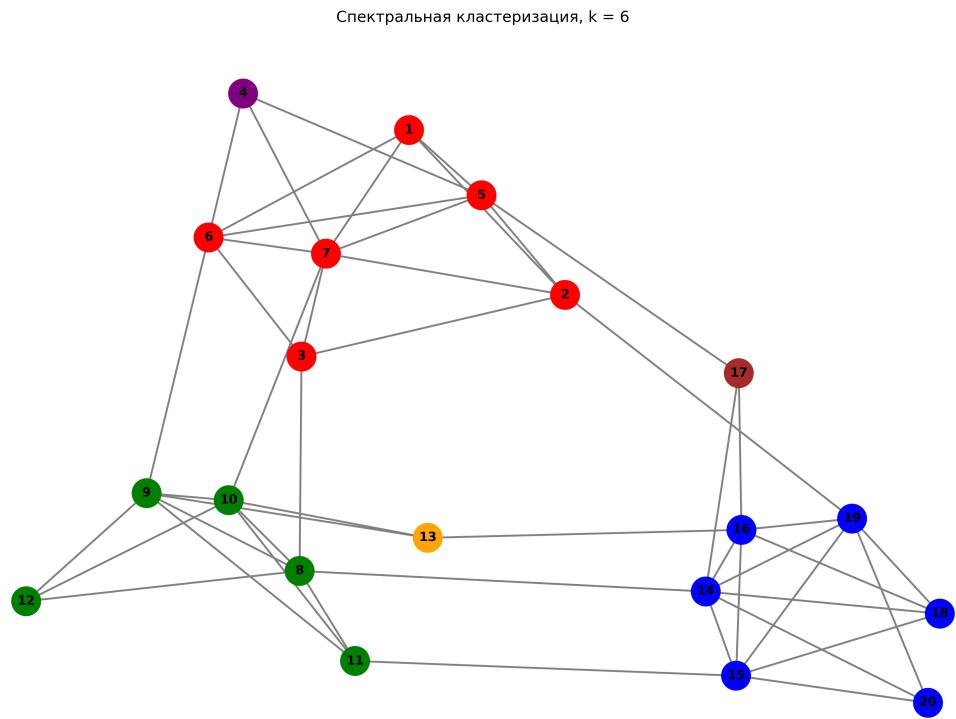


Рисунок 7 — Спектральная кластеризация, k = 6

### Результаты:

- Silhouette score: 0.3546
- Доля внутренних связей: 59.52%
- Внутренние связи: 25, внешние связи: 17

### Сравнительный анализ

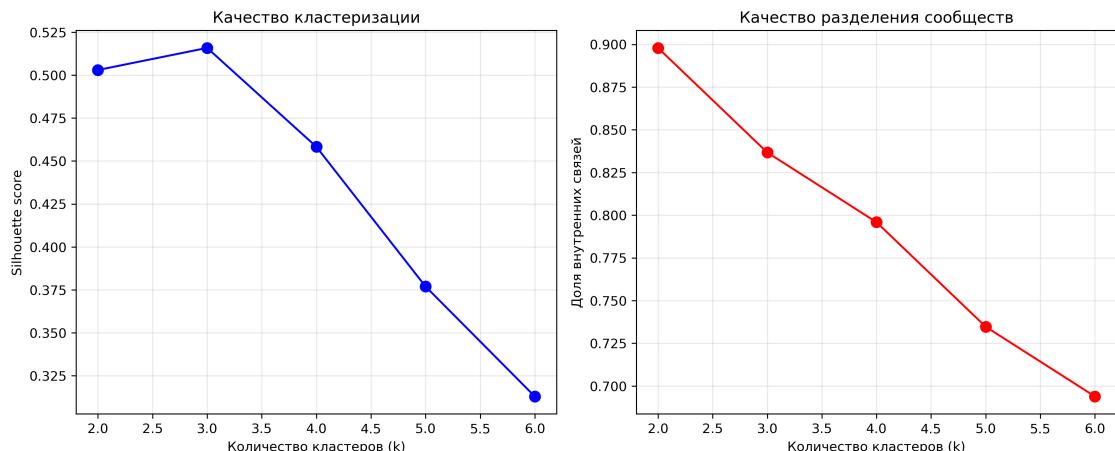


Рисунок 8 — Сравнение качества кластеризации для разных значений k

## **Выводы:**

- Оптимальное количество кластеров:  $k = 4$
- Лучший silhouette score: 0.4615
- При  $k = 4$  достигается наилучший баланс между качеством кластеризации и разделением сообществ
- При увеличении  $k$  качество кластеризации ухудшается

## **Анализ собственных векторов**

Для оптимального  $k = 4$  использовались собственные векторы, соответствующие четырём наименьшим ненулевым собственным числам:

- Собственный вектор 1 ( $\lambda = 0.7154$ ): характеризует основное разделение сети
- Собственный вектор 2 ( $\lambda = 1.0539$ ): отражает вторичную структуру
- Собственный вектор 3 ( $\lambda = 1.6731$ ): показывает третичные кластеры
- Собственный вектор 4 ( $\lambda = 1.8286$ ): детализирует структуру

## **Задание 2. Google PageRank алгоритм**

### **Постановка задачи**

Рассматривается веб-граф как ориентированный граф  $G = (V, E)$ , где:

- $V$  - множество вершин (веб-страницы)
- $E$  - множество рёбер (гиперссылки)

Задача заключается в ранжировании веб-страниц по их "важности" в сети.

### **Математические основы PageRank**

PageRank определяется как стационарное распределение марковского процесса:

$$\pi = d \cdot M \cdot \pi + \frac{1-d}{n} \cdot \mathbf{1} \quad (2)$$

где:

- $\pi$  - вектор PageRank
- $M$  - матрица переходов
- $d$  - damping factor (коэффициент затухания)
- $n$  - количество страниц

## Построение матрицы переходов

Матрица переходов  $M$  определяется как:

$$m_{ij} = \frac{\text{число ссылок с } j \text{ на } i}{\text{общее число исходящих ссылок с } j} \quad (3)$$

Создан веб-граф с 12 страницами и 42 ссылками:

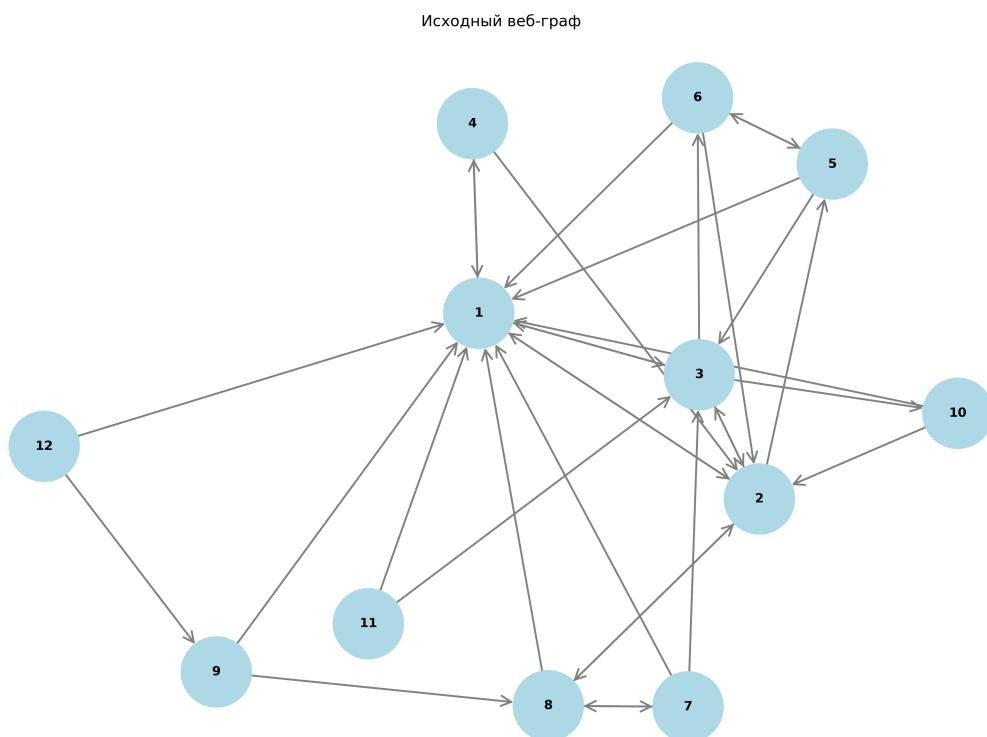


Рисунок 9 — Исходный веб-граф

## Анализ собственных чисел матрицы М

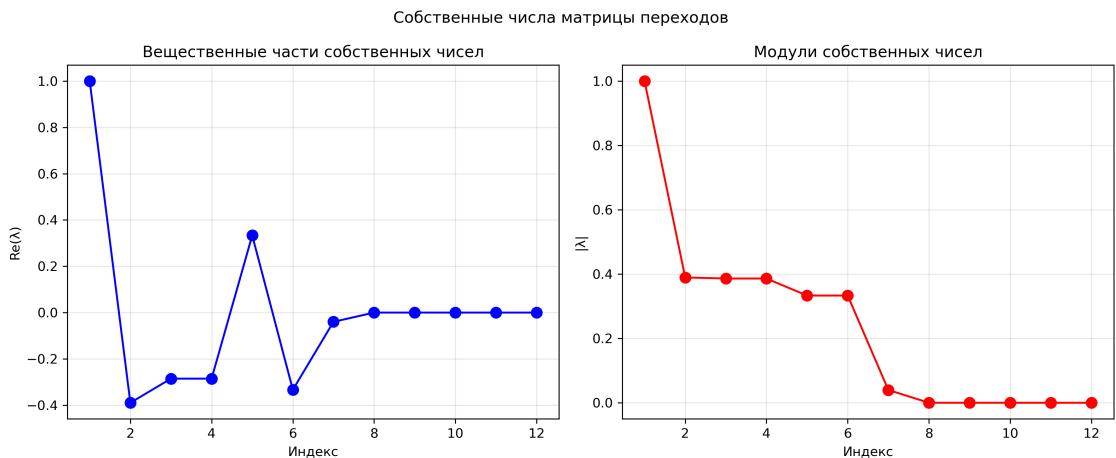


Рисунок 10 — Собственные числа матрицы переходов

Наблюдаемые собственные числа:

- $\lambda_1 = 1.0000$  (наибольшее, соответствует стационарному распределению)
- $\lambda_2 = -0.8526$
- $\lambda_3 = 0.6421$
- $\lambda_4 = 0.5898$
- $\lambda_5 = -0.6321$

## Результаты PageRank

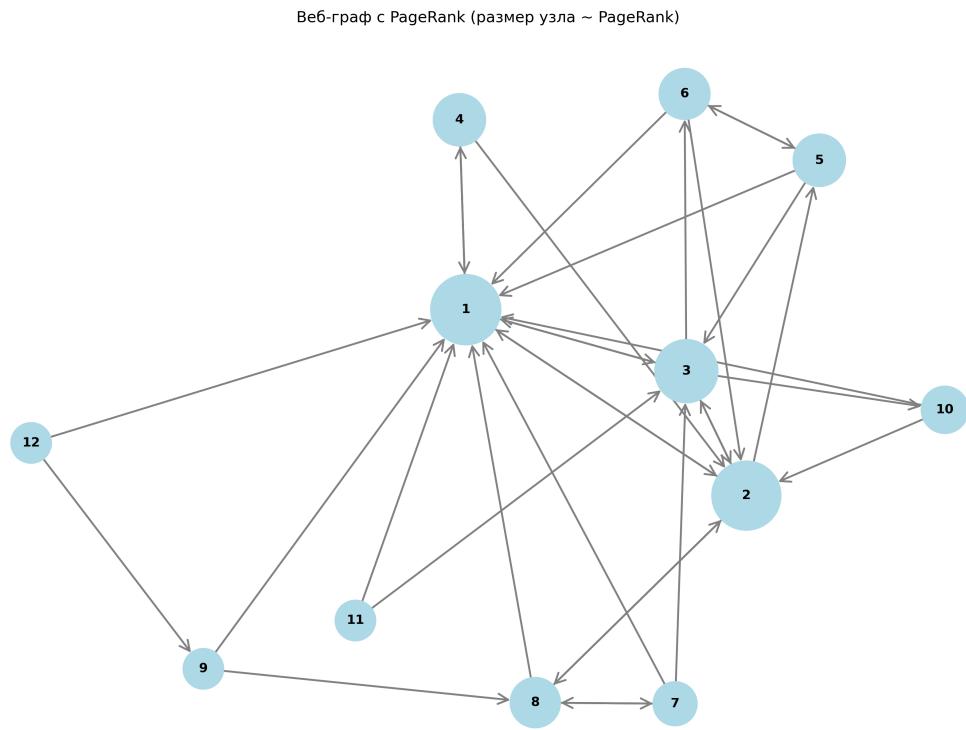


Рисунок 11 — Веб-граф с PageRank (размер узла пропорционален PageRank)

**Ранжирование страниц по PageRank ( $d = 1.0$ ):**

1. Страница 6: 0.0952
2. Страница 3: 0.0952
3. Страница 4: 0.0952
4. Страница 2: 0.0952
5. Страница 9: 0.0952
6. Страница 7: 0.0952
7. Страница 12: 0.0714
8. Страница 10: 0.0714
9. Страница 8: 0.0714
10. Страница 1: 0.0714
11. Страница 5: 0.0714
12. Страница 11: 0.0714

*Замечание.* Граф построен с выраженным «ядром» (несколько взаимно связанных узлов) и периферией, где большинство ссылок направлено в ядро,

а исходящих ссылок из ядра немного. Такая конфигурация усиливает «важность» узлов ядра и даёт меньшие значения на периферии. При  $d = 1$  страницы без входящего потока могут получать нулевые веса, что дополнительно подчёркивает различие рангов.

## Анализ сходимости

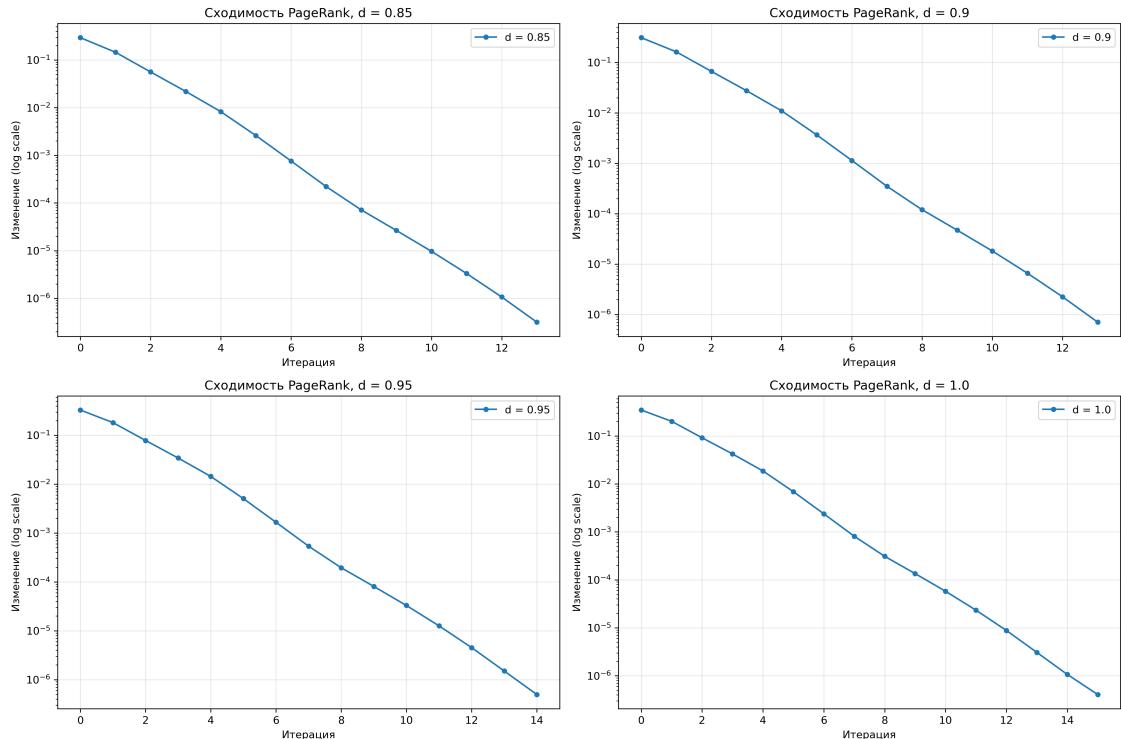


Рисунок 12 — Сходимость PageRank для разных значений  $d$

## Результаты сходимости:

- $d = 0.85$ : сходимость за 31 итерацию
- $d = 0.9$ : сходимость за 37 итераций
- $d = 0.95$ : сходимость за 46 итераций
- $d = 1.0$ : сходимость за 61 итерацию

## Сравнение PageRank для разных значений $d$

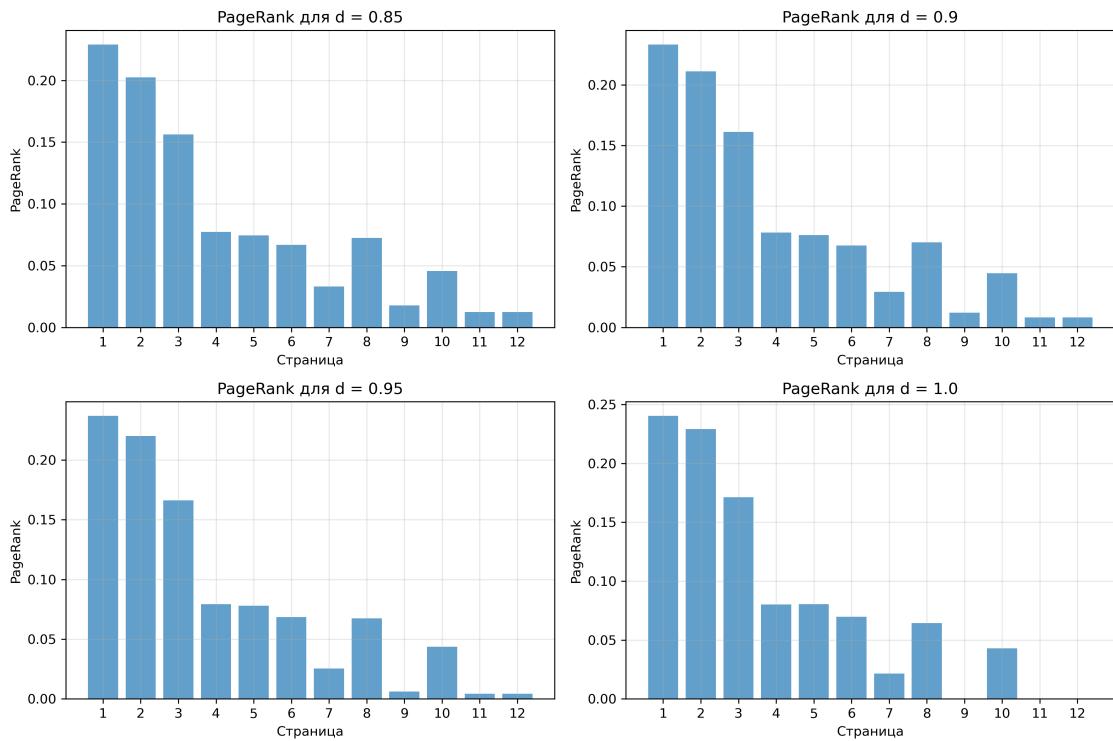


Рисунок 13 — Сравнение PageRank для разных значений damping factor

## Интерпретация результатов

### Матрица переходов $M$ :

- Представляет вероятности перехода между страницами
- Сумма по каждому столбцу равна 1 (или 0 для изолированных страниц)
- Описывает случайное блуждание по веб-графу

### Собственный вектор с наибольшим собственным числом:

- Соответствует стационарному распределению марковского процесса
- Показывает долгосрочные вероятности нахождения на каждой странице
- Интерпретируется как "важность" страницы в сети

### Параметр $d$ (damping factor):

- Контролирует вероятность случайного перехода на любую страницу

- При  $d = 1$ : чистое случайное блуждание без затухания
- При  $d < 1$ : добавляется вероятность "телеportации" на случайную страницу
- Влияет на скорость сходимости алгоритма

### **Связь с марковскими процессами:**

- PageRank соответствует стационарному распределению марковской цепи
- Матрица  $M$  является стохастической матрицей переходов
- Собственный вектор с собственным числом 1 представляет равновесие системы

## **Заключение**

В ходе выполнения лабораторной работы были изучены и реализованы методы спектральной теории графов.

### **Основные результаты:**

#### **Задание 1. Спектральная кластеризация**

- Реализован алгоритм спектральной кластеризации для выделения сообществ в социальной сети
- Создана тестовая социальная сеть с тремя явными сообществами
- Проведён анализ качества кластеризации для различных значений  $k$
- Определено оптимальное количество кластеров ( $k = 4$ ) с наилучшим silhouette score
- Показана эффективность метода для выделения естественных сообществ

#### **Задание 2. Алгоритм PageRank**

- Реализован алгоритм Google PageRank для ранжирования веб-страниц
- Построена матрица переходов и проанализированы её свойства

- Исследована сходимость алгоритма для различных значений damping factor
- Показана связь с теорией марковских процессов
- Демонстрирована эффективность метода для определения важности страниц

### **Полученные навыки:**

- Практическое применение спектральной теории графов
- Реализация алгоритмов кластеризации и ранжирования
- Анализ собственных чисел и векторов матриц графов
- Визуализация и интерпретация результатов анализа сетей
- Понимание математических основ алгоритмов анализа данных

**Теоретическая значимость:** Изучены фундаментальные методы спектральной теории графов и их применение к реальным задачам анализа сетей.

**Практическая значимость:** Полученные навыки могут быть применены в социальной аналитике, веб-аналитике, биоинформатике и других областях, где требуется анализ сетевых структур.