

به نام خداوند بخشنده مهربان

## پروژه پایانی جبر خطی

حمیدرضا ربیعی، مریم رضائی

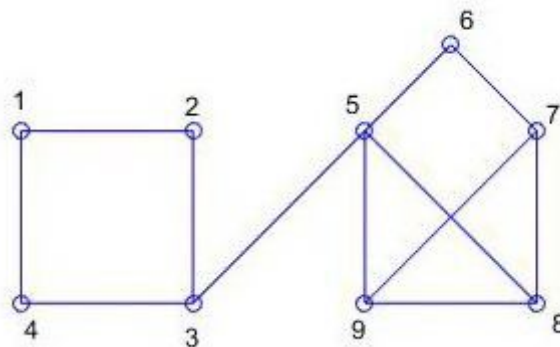
نام و نام خانوادگی:

شماره دانشجویی:

### خوشه‌بندی کاربران در شبکه‌های اجتماعی

هدف از این پروژه، جدا کردن و گروه‌بندی رئوس یک گراف همبند به صورت خوشه‌هایی است که بیشترین ارتباط را با هم دارند. برای مثال، ارتباط دوستی در یک شبکه‌ی اجتماعی را به صورت یال‌های بدون جهت بین کاربران در نظر بگیرید. می‌خواهیم کاربرانی که با هم بیشترین ارتباط را داشته و اجتماع‌های کوچک را تشکیل می‌دهند، پیدا کنیم. مثلاً در گراف زیر، قرار است رئوس‌های ۱ تا ۴ در یک دسته و رئوس‌های ۵ تا ۹ در دسته‌ی دیگر قرار گیرند.

در این پروژه، ابتدا با دنبال کردن مراحل، قدم به قدم گراف زیر را خوشه‌بندی می‌کنید، سپس خودتان مراحل را روی گراف شامل ۳۵۱ فرد اجرا می‌کنید تا خوشه‌هایشان مشخص شود.



```
In [ ]: import numpy as np
import matplotlib.pyplot as plt
import networkx as nx
```

```
In [ ]: adj = np.array()
```

```
In [ ]: def show_graph(adjacency_matrix, labels=None):
    color_map = {1: 'blue', 2: 'green', 3: 'red', 4: 'yellow'}
    colors = [color_map[x] for x in labels] if labels is not None else None

    rows, cols = np.where(adjacency_matrix == 1)
    edges = zip(rows.tolist(), cols.tolist())
    gr = nx.Graph()
    gr.add_edges_from(edges)
    nx.draw(gr, node_size=500, node_color=colors)
    plt.show()

show_graph(adj, [1, 1, 1, 1, 2, 2, 2, 2, 2])
```

می‌توانید از تابع `show_graph` برای نمایش گراف خود استفاده کنید. ورودی اول این تابع ماتریس مجاورت بوده و ورودی دوم آن در صورت داده شدن، باید به صورت یک لیست از اعداد ۱ تا ۴ باشد. بعد از خوشه‌بندی می‌توانید شماره‌ی خوشه‌ی متناظر با هر راس را در این لیست قرار دهید تا رنگ آن‌ها مشخص باشند.

## گراف تمرینی

ابتدا ماتریس مجاورت گراف را به شکل یک ماتریس در `numpy` تعریف کنید. در این ماتریس، اگر بین دو راس `i` و `j` یال وجود داشته باشد، درایه‌ی `i,j`

یک خواهد بود. بنابراین به دلیل بدون جهت بودن گراف، ماتریس متقارن خواهد بود.

In [ ]: # Your code

در این مرحله، ابتدا جمع سطرها را محاسبه کرده و در بردار  $Sum$  ذخیره کنید. سپس ماتریس لاپلاسیان را به صورت  $Lapl = diag(Sum) - Adj$  محاسبه کنید. منظور از  $diag(x)$  ماتریس قطری‌ای است که درایه‌های  $x$  روی قطر اصلی آن قرار گرفته و بقیه‌ی درایه‌های آن صفر هستند. هدف از محاسبه‌ی این ماتریس این است که با حفظ ویژگی‌های ماتریس مجاورت، جمع سطرها و ستون‌های آن نیز صفر شوند.

In [ ]: # Your code

حال باید مقدار ویژه و بردار ویژه‌های ماتریس را استخراج کنید. همچنین باید این مقادیر ویژه و بردار ویژه‌های متناظرشان، از کوچک به بزرگ مرتب شده باشند.

In [ ]: # Your code

چون جمع سطرهای ماتریس لاپلاسیان صفر بود، ماتریس تکین بوده و یک مقدار ویژه‌ی صفر خواهد داشت. همچنین ماتریس مثبت نیمه معین است و همه‌ی مقادیر ویژه نامنفی هستند. بنابراین کوچک‌ترین مقدار ویژه صفر بوده و مقدار ویژه‌ی بعدی مثبت خواهد بود. بردار ویژه‌ی متناظر با این مقدار ویژه را در نظر بگیرید. این بردار ویژه متناظر با کمترین واریانس است. (برای اطلاعات بیشتر در مورد PCA مطالعه کنید.) بنابراین مناسب‌ترین معیار برای خوشه‌بندی است. راس‌هایی که درایه‌ی متناظرشان در این بردار ویژه مثبت است را در یک خوشه و راس‌هایی که منفی است را در خوشه‌ی دیگر در نظر بگیرید. با بررسی خروجی می‌توانید تبدیل شدن گراف اولیه به دو خوشه‌ی گفته شده در ابتدای پروژه را مشاهده کنید.

In [ ]: # Your code

گراف را با خوشه‌هایی که بدست آوردید توسط تابع `show_graph` نمایش دهید.

In [ ]: `# Your code`

برای داشتن تعداد بیشتری خوشه، می‌توان بردار ویژه‌های بیشتری در نظر گرفت. برای مثال، برای داشتن ۴ خوشه می‌توانید بردار ویژه‌ی متناظر با ۲ کوچکترین مقدار ویژه‌های مثبت را در نظر بگیرید. حال به ازای هر راس، دو درایه در این دو بردار ویژه داریم. می‌توانید ۴ خوشه را به صورت زیر در نظر بگیرید:

•  $(+, +)$

•  $(+, -)$

•  $(-, +)$

•  $(-, -)$

با انجام این خوشه‌بندی روی گراف اولیه، می‌توانید مشاهده کنید خوشه‌بندی مناسب نبوده و همان دو خوشه‌ی اولیه دقیق‌تر بودند.

In [ ]: `# Your code`

گراف را با خوشه‌هایی که بدست آوردید توسط تابع `show_graph` نمایش دهید.

In [ ]: `# Your code`

## گراف اصلی

اگر چه در گراف تمرینی خوشه‌بندی با چشم هم قابل مشاهده بود، در گراف با رئوس بیشتر این کار به صورت چشمی ممکن نیست. در این قسمت، مراحل بالا را روی گراف اصلی اجرا کنید و خوشه‌بندی با ۲ و ۴ خوشه را بدست آورید.

```
In [ ]: adj_final = np.zeros((100, 100))
        file1 = open('data.txt', 'r')
        lines = file1.readlines()
        print(len(lines))
        for l in lines:
            i, j = l.split()
            adj_final[int(i) - 1, int(j) - 1] = 1
            adj_final[int(j) - 1, int(i) - 1] = 1
```

```
In [ ]: # Your code
```

گراف‌هایی که با ۲ و ۴ خوشه، خوشه‌بندی کردید را توسط تابع `show_graph` نمایش دهید.

```
In [ ]: # Your code
```