



Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет имени
Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехники и комплексной автоматизации»
КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ по дисциплине «Вычислительная математика»

| | |
|--------------|---------------------------------------|
| Студент: | Караф Сармат Майк |
| Группа: | РК6-52Б |
| Тип задания: | лабораторная работа |
| Тема: | Спектральное и сингулярное разложение |

Студент

подпись, дата

Караф С. М.

Фамилия, И.О.

Преподаватель

подпись, дата

Соколов А. П.

Фамилия, И.О.

Москва, 2021

Содержание

| | |
|---|----------|
| Спектральное и сингулярное разложения | 3 |
| 1 Задание | 3 |
| 2 Цель выполнения лабораторной работы | 5 |
| 3 Выполненные задачи | 5 |
| 4 Базовая часть | 6 |
| 4.1 Реализация функции $pca(A)$, которая возвращает список главных компонент и список соответствующих стандартных отклонений. . . . | 6 |
| 4.2 Набор данных Breast Cancer Wisconsin Data set. | 7 |
| 4.3 Нахождение главных компонентов из указанного набора данных. . . | 8 |
| 4.4 Визуализация графа стандартного отклонения. | 8 |
| 4.4 Визуализация сепарации типов опухолей по первым двум компонентам. | 9 |
| 5 Продвинутая часть. | 10 |
| 5.1 Построение лапласиан для трех графов. | 10 |
| 5.2 Доказательство, что невзвешенный граф с n вершинами является положительно полуопределенной матрицей, имеющей n неотрицательных собственных чисел, одно из которых равно нулю. . . . | 12 |
| 5.3 Нахождение спектра для каждого графа. Выявления особенностей спектра и связь с количеством кластеров. | 14 |
| 5.4 Нахождение количество кластеров для третьего графа. Визуализация графа исходной матрицы смежности и её отсортированной версии. | 17 |
| 6 Заключение | 18 |

Спектральное и сингулярное разложения

1 Задание

Спектральное разложение (разложение на собственные числа и вектора) и сингулярное разложение, то есть обобщение первого на прямоугольные матрицы, играют настолько важную роль в прикладной линейной алгебре, что тяжело придумать область, где одновременно используются матрицы и не используются указанные разложения в том или ином контексте. В базовой части лабораторной работы мы рассмотрим метод главных компонент (англ. Principal Component Analysis, *PCA*), без преувеличения самый популярный метод для понижения размерности данных, основой которого является сингулярное разложение. В продвинутой части мы рассмотрим куда менее очевидное применение разложений, а именно одну из классических задач спектральной теории графов – задачу разделения графа на сильно связанные компоненты (кластеризация на графе).

Требуется (базовая часть).

1. Написать функцию $pca(A)$, принимающую на вход прямоугольную матрицу данных A и возвращающую список главных компонент и список соответствующих стандартных отклонений.
2. Скачать набор данных Breast Cancer Wisconsin Data set: <https://archrk6.bmstu.ru/index.php/f/85484391>.
Указанный датасет хранит данные 569 пациентов с опухолью, которых обследовали на предмет наличия рака молочной железы. В каждом обследовании опухоль была проклассифицирована экспертами как доброкачественная (benign, 357 пациентов) или злокачественная (malignant, 212 пациентов) на основе детального исследования снимков и анализов. Дополнительно на основе снимков был автоматически выявлен и задокументирован ряд характеристик опухолей: радиус, площадь, фрактальная размерность и так далее (всего 30 характеристик). Постановку диагноза можно автоматизировать, если удастся создать алгоритм, классифицирующий опухоли исключительно на основе этих автоматически получаемых характеристик. Указанный файл является таблицей, где отдельная строка соответствует отдельному пациенту. Первый элемент в строке обозначает ID пациента, второй элемент – диагноз (M = malignant, B = benign), и оставшиеся 30 элемент соответствуют характеристикам опухоли (их детальное описание находится в файле <https://archrk6.bmstu.ru/index.php/f/854842>).
3. Найти главные компоненты указанного набора данных, используя функцию $pca(A)$.
4. Вывести на экран стандартные отклонения, соответствующие номерам главных компонент.

✓

5. Продемонстрировать, что проекций на первые две главные компоненты достаточно для того, чтобы произвести сепарацию типов опухолей (доброкачественная/злокачественная) для подавляющего их большинства. Для этого необходимо вывести на экран проекции каждой из точек, используя scatter plot.

Требуется (продвинутая часть):

1. Построить лапласианы (матрицы Кирхгофа) L для трех графов:
 - полный граф G_1 , имеющий 10 узлов;
 - граф G_2 , изображенный на рисунке 1;
 - граф G_3 , матрица смежности которого хранится в файле <https://archrk6.bms.tu.ru/index.php/f/854844>,

где лапласианом графа называется матрица $L = D - A$, где A – матрица смежности D – матрица, на главной диагонали которой расположены степени вершин графа, а остальные элементы равны нулю.

2. Доказать, что лапласиан неориентированного невзвешенного графа с n вершинами является положительно полуопределенной матрицей, имеющей n неотрицательных собственных чисел, одно из которых равно нулю.
3. Найти спектр каждого из указанных графов, т.е. найти собственные числа и вектора их лапласианов. Какие особенности спектра каждого из графов вы можете выделить? Какова их связь с количеством кластеров.
4. Найти количество кластеров в графе G_3 , используя второй собственный вектор лапласиана. Для демонстрации кластеров выведите на графике исходную матрицу смежности и ее отсортированную версию.

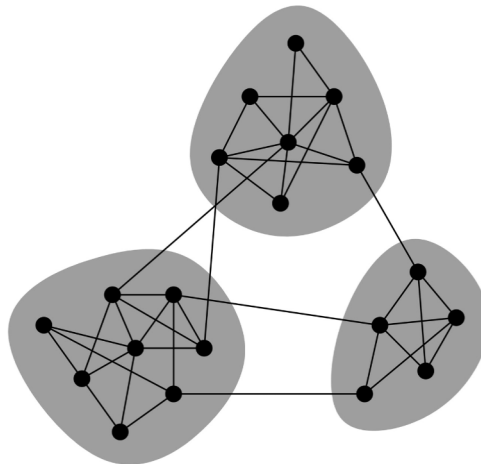


Рис. 1. Граф, содержащий три кластера.

2 Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – рассмотреть спектральное и сингулярное разложение, используя метод нахождения главного компонента и рассмотрев классическую задачу спектральной теории графов.

3 Выполненные задачи

Базовая часть:

1. Реализована функция $pca(A)$, которая принимает на вход прямоугольную матрицу данных A и возвращает список главных компонент и список соответствующих стандартных отклонений.
2. Подключён набор данных Breast Cancer Wisconsin Data set к данной работе.
3. Найдены главные компоненты указанного набора данных, используя функцию $pca(A)$.
4. Выведены стандартные отклонения, которые соответствуют номерам главных компонент.
5. Продемонстрирована сепарация типов опухолей.

Продвинутая часть:

1. Построены лапласианы L для трех графов:
 - полный гра G_1 , имеющий 10 узлов;
 - граф G_2 , изображенный на рисунке 1;
 - граф G_3 , матрица смежности из установленного файла.
2. Доказано, что лапласиан неориентированного невзвешенного графа с n вершинами является положительно полуопределенной матрицей, имеющей n неотрицательных собственных чисел, одно из которых равно нулю.
3. Найдены спектры каждого из указанных графов. Были выведены особенности спектра каждого графа и связь с количеством кластеров.
4. Найдено количество кластеров в графе G_3 , используя второй собственный вектор лапласиана. Визуализация исходной матрицы смежности и ее отсортированной версии.

4 Базовая часть

Лабораторная работа выполнена на языке программирования Python с использованием библиотек: *numpy*, *matplotlib*, *pandas* и *sklearn*.

4.1 Реализация функции $pca(A)$, которая возвращает список главных компонент и список соответствующих стандартных отклонений.

В данной работе, для реализации базовой части необходимо рассмотреть метод главных компонент, для написания функции $pca(A)$. Рассмотрение ковариационной матрицы $K \in \mathbb{R}^{n \times n}$.

$$K = \nu A^T A \quad (1)$$

Коэффициент ν - зависит от выбора оценки ковариации. Элементы этой матрицы k_{ij} являются ковариациями i -х и j -х показаний. Ковариационная матрица при этом является матрицей Грамма, т.е. положительно полуопределенной.

$$k_{ij} = \sigma_{x_i x_j} = \nu \sum_{k=1}^m (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j) = \nu \alpha_i \cdot \alpha_j$$

Так же для нахождения главного компонента необходимо вычислить сингулярные числа и вектора.

Сингулярными числами ν_1, \dots, ν_r матрицы $A \in \mathbb{R}^{n \times n}$ называются числа $\nu_i = \sqrt{\lambda_i}$, где λ_i являются ненулевыми собственными числами соответствующей матрицы Грамма $K = A^T A$. Ассоциированные собственные вектора матрицы Грамма K называются сингулярными векторами. Сингулярное разложение матрицы $A \in \mathbb{R}^{n \times n}$ ранга $r > 0$ может быть представлено в виде разложения

$$A = P \Sigma Q^T,$$

где $P \in \mathbb{R}^{n \times n}$ и $Q \in \mathbb{R}^{n \times n}$ - ортонормальные матрицы ($P^T P = E$ и $Q^T Q = E$), а $\Sigma \in \mathbb{R}^{n \times n}$ - диагональная матрица, на диагонали которой расположены сингулярные числа ν_1, \dots, ν_r .

Главными компонентами, матрицы центрированных данных A , являются её сингулярные вектора, при этом j -я главная компонента соответствует j -му сингулярному вектору q_j и стандартному отклонению

$$\sqrt{\nu} \sigma_j, \quad (2)$$

где σ_j является j -м сингулярным числом, $\nu = \frac{1}{m-1}$.

Функция $pca(A)$ (листинг 1), возвращает список главных компонент и список соответствующих стандартных отклонений. Она принимает на вход матрицу центрированных данных A . При помощи функции *np.linalg.eig()* вычисляет собственные числа, вектор матрицы Грамма 1, и находит сингулярные числа. Найденные сингулярные числа сортируются по убыванию, в соответствии с сингулярными числами. Стандартное отклонение вычисляется по формуле 2.

Листинг 1. Программная реализация функции $pca(A)$, для нахождения главного компонента и стандартных отклонений.

```

1 def pca(A):
2     W, V = np.linalg.eig(A.T @ A)
3     ind = np.flip(np.argsort(np.sqrt(W)))
4     sigma = np.sqrt(W)[ind]
5     self_vec = V[:, ind]
6     return self_vec.T, np.sqrt(1 / (A.shape[0] - 1)) * sigma

```

4.2 Набор данных Breast Cancer Wisconsin Data set.

Используется набор данных Breast Cancer Wisconsin Data set, для рассмотрения метода главных компонент. Данные предоставлены с сайта [https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

Указанный дата сет хранит данные 569 пациентов с опухолью, которых обследовали на предмет наличия рака молочной железы. В каждом обследовании опухоль была проклассифицирована экспертами как доброкачественная (benign, 357 пациентов) или злокачественная (malignant, 212 пациентов) на основе детального исследования снимков и анализов. Дополнительно на основе снимков был автоматически выявлен и задокументирован ряд характеристик опухолей (всего 30 характеристик) (Рис.2)

| | id | diagnosis | radius_mean | texture_mean | perimeter_mean | ... | compactness_worst | concavity_worst | concave | points_worst | symmetry_worst | fractal_dimension_worst |
|-----|----------|-----------|-------------|--------------|----------------|-----|-------------------|-----------------|---------|--------------|----------------|-------------------------|
| 0 | 842302 | M | 17.99 | 10.38 | 122.80 | ... | 0.66560 | 0.7119 | | 0.2654 | 0.4601 | 0.11890 |
| 1 | 842517 | M | 20.57 | 17.77 | 132.90 | ... | 0.18660 | 0.2416 | | 0.1860 | 0.2750 | 0.08902 |
| 2 | 84300903 | M | 19.69 | 21.25 | 130.00 | ... | 0.42450 | 0.4504 | | 0.2430 | 0.3613 | 0.08758 |
| 3 | 84348301 | M | 11.42 | 20.38 | 77.58 | ... | 0.86630 | 0.6869 | | 0.2575 | 0.6638 | 0.17300 |
| 4 | 84358402 | M | 20.29 | 14.34 | 135.10 | ... | 0.20500 | 0.4000 | | 0.1625 | 0.2364 | 0.07678 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | | ... | ... | ... |
| 564 | 926424 | M | 21.56 | 22.39 | 142.00 | ... | 0.21130 | 0.4107 | | 0.2216 | 0.2060 | 0.07115 |
| 565 | 926682 | M | 20.13 | 28.25 | 131.20 | ... | 0.19220 | 0.3215 | | 0.1628 | 0.2572 | 0.06637 |
| 566 | 926954 | M | 16.60 | 28.08 | 108.30 | ... | 0.30940 | 0.3403 | | 0.1418 | 0.2218 | 0.07820 |
| 567 | 927241 | M | 20.60 | 29.33 | 140.10 | ... | 0.86810 | 0.9387 | | 0.2650 | 0.4087 | 0.12400 |
| 568 | 92751 | B | 7.76 | 24.54 | 47.92 | ... | 0.06444 | 0.0000 | | 0.0000 | 0.2871 | 0.07039 |

[569 rows x 32 columns]

Рис. 2. Breast Cancer Wisconsin Data Set

Считывание данных реализованно при помощи библиотеки *pandas*. Для дальнейшей работы с этими данными нужно их привести в удобную форму, для этого производится отбор нужных для работы данных. Так же необходимо пересчитать данные по формуле 3 для удобного масштабирования. Операция масштабирования выполняется при помощи функции `StandardScaler()` из библиотеки `sklearn.preprocessing`.

$$z = \frac{x - \mu}{\sigma}, \quad (3)$$

где μ - среднее значение ($\mu = \frac{1}{N} \sum_{i=1}^m x_i$) и σ - стандартное отклонение ($\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^m (x_i - \mu)^2}$)
Процесс преобразование данных показан в листинге 2.

Листинг 2. Приведение данных в удобную форму для работы.

```
1 data = pd.read_csv("../Data_set/Breast Cancer Wisconsin.csv", header=0)
2 diagnosis = np.array(data.loc[:, ["diagnosis"]])
3 only_data = np.array(data.iloc[:, 2:])
4 scale = StandardScaler()
5 scale.fit(only_data)
6 only_data = scale.transform(only_data)
```

4.3 Нахождение главных компонент из указанного набора данных.

Для нахождения главных компонент из матрицы данных необходимо центрировать матрицу, данная операция будет выполняться в реализованной функции *get_normalized_data_matrix(Matrix)*.

Рассматриваются отклонения данных от средних значений, т.е. $a_{ij} = x_{ij} - \bar{x}_j$. Это дает матрицу центрированных матриц A_c :

$$A_c = X - e\bar{x} = \left(E - \frac{1}{m}ee^T\right)X, \quad (4)$$

где e - единичный вектор, ee^T - матрица единиц, $\bar{x} = \frac{1}{m}e^T X$ - вектор выборочных средних показаний измерений, m - количество строк в матрице, E - матрица в котором главная диагональ заполнена единицами, а остальные нулями.

Функция *get_normalized_data_matrix(Matrix)* (листинг 3), возвращает центрированную матрицу. Она принимает на вход матрицу масштабированных данных. При помощи формулы (4) преобразует ее в центрированную матрицу.

Листинг 3. Функция преобразования матрицы масштабированных данных в центрированную матрицу.

```
1 def get_normalized_data_matrix(Matrix):
2     m = Matrix.shape[0]
3     A = (np.eye(m) - 1. / m * np.ones((m, m))) @ Matrix
4     return A
```

После получения центрированной матрицы, она передается в функцию *pca(a)*, которая продемонстрирована в листинге (1), в котором вычисляются главные компоненты.

4.4 Визуализация графа стандартного отклонения.

Из полученных ранее данных можно выявить зависимость стандартного отклонения (2), от индекса главного компонента. Данная зависимость представлена на графике (3) и в листинге (4)

Листинг 4. Алгоритм визуализации графа стандартного отклонения.

```
1 plt.subplots(figsize=(10, 6))
2 plt.plot(1. + np.arange(len(Vector)), Vector, "o—", color="grey")
3 plt.xlabel('index', fontsize=20)
4 plt.ylabel(r'Standard Deviation', fontsize=20)
5 plt.title("Standard Deviation From Component Number", fontsize=20)
6 plt.grid()
7 plt.show()
```

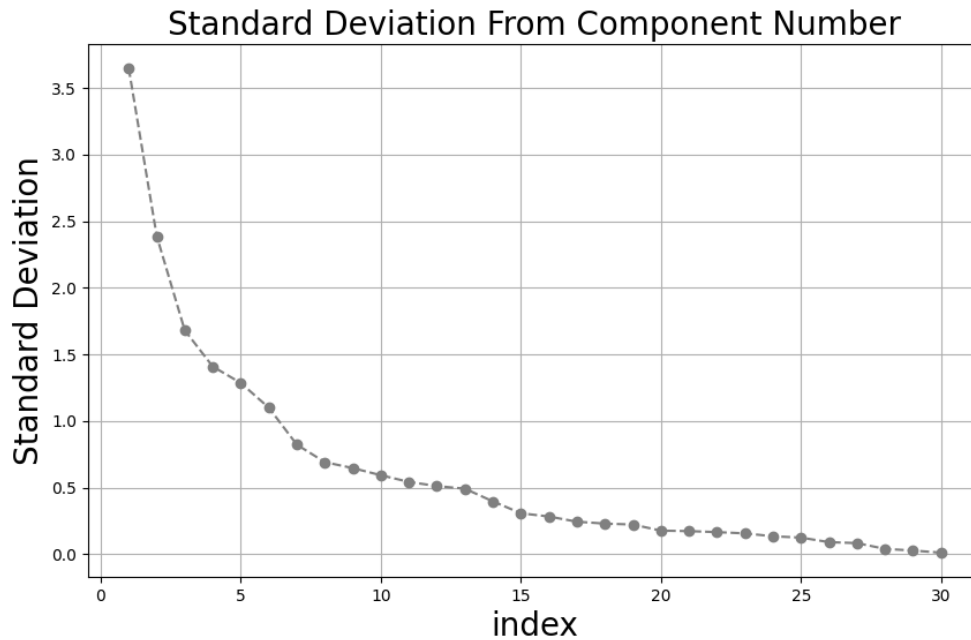


Рис. 3. Зависимость стандартного отклонения от индекса компонента.

На графике (Рис. 3) можно заметить, что наибольшее стандартное отклонение имеют первые два компонента. Можно сделать вывод что из всех предоставленных компонент первые хранят большую информацию, значит в дальнейшем будут рассматриваться только первые два компонента.

4.4 Визуализация сепарации типов опухолей по первым двум компонентам.

Из полученных ранее данных, можно визуализировать сепарацию опухолей, спроецировав их на главные компоненты.

$$A_k = A \times Q^T, \quad (5)$$

где M - матрица центрирования, Q - матрица главных компонентов. Визуализация сепарации типов опухолей по первым двум компонентам представлено на графе (Рис.4) и алгоритм представлен в листинге 5.

Листинг 5. Алгоритм визуализации графа сепарации типов опухолей.

```
1 plt.subplots(figsize=(14, 6))
2 plt.xlabel('Principal Component - 1', fontsize=20)
3 plt.ylabel('Principal Component - 2', fontsize=20)
4 plt.title("Principal Component Analysis of Breast Cancer Dataset", fontsize=20)
5 for i in range(len(only_data)):
6     if diagnosis[i][0] == "M":
7         grey_dot = plt.scatter(Matrix_k[i][0], Matrix_k[i][1], color="grey")
8     else:
9         orange_dot = plt.scatter(Matrix_k[i][0], Matrix_k[i][1], color="orange")
10 plt.legend([grey_dot, orange_dot], ['Malignant', 'Benign'])
```

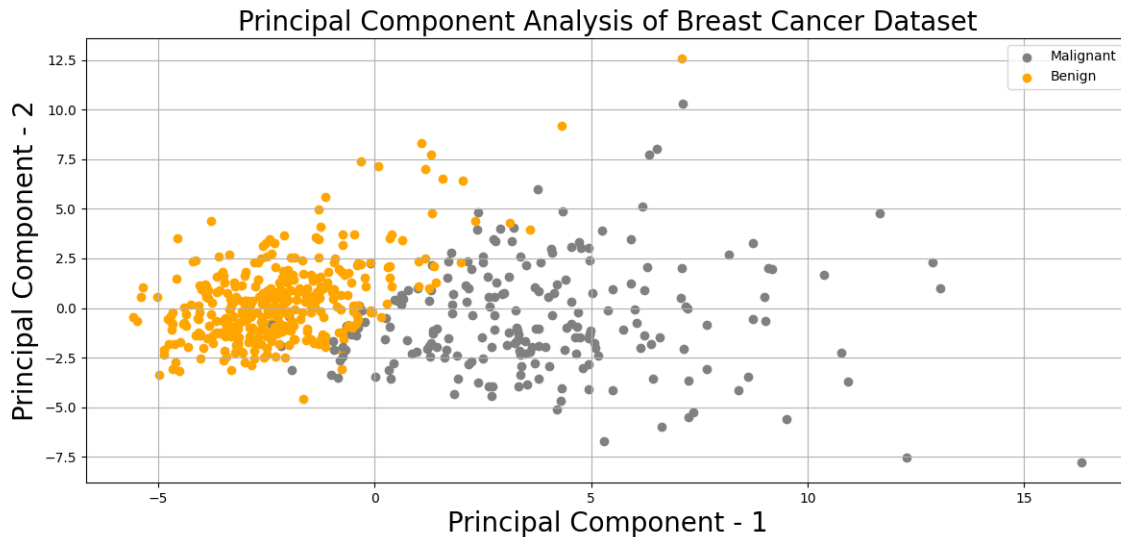


Рис. 4. Сепарации типов опухолей по первым двум компонентам.

На графике (Рис.4) видны злокачественные и доброкачественные опухоли, следовательно двух компонентов достаточно для сепарации данных.

5 Продвинутая часть.

В продвинутой части происходит рассмотрение классической задачи спектральной теории графов – задачу разделения графа на сильно связанные компоненты (кластеризация на графе).

5.1 Построение лапласиан для трех графов.

Для построения лапласиан для трех графов G_1, G_2, G_3 потребовалось написать функцию $Laplas(Matrix)$, которая реализует данное построение.

Лапласиан графа - это матрица:

$$L = D - A, \quad (6)$$

где D - матрица, на главной диагонали которой расположены степени вершин графа, а остальные элементы равны нулю, A - матрица смежности.

Графы G_1, G_2, G_3 имеют следующий вид:

Полный граф G_1 имеет 10 узлов. Матрица смежности A_1 представлена на рисунке 5.

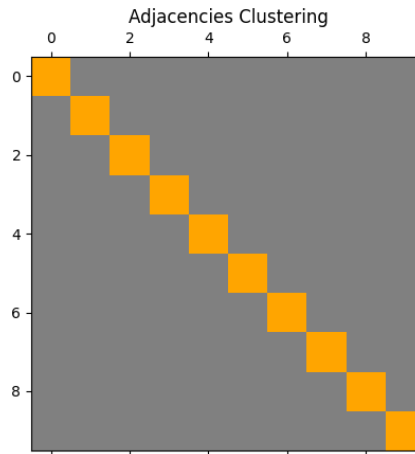


Рис. 5. Матрица смежности графа G_1 .

Граф G_2 изображенный на рисунке 1 имеет 20 вершин. Матрица смежности A_2 представлена на рисунке 6.

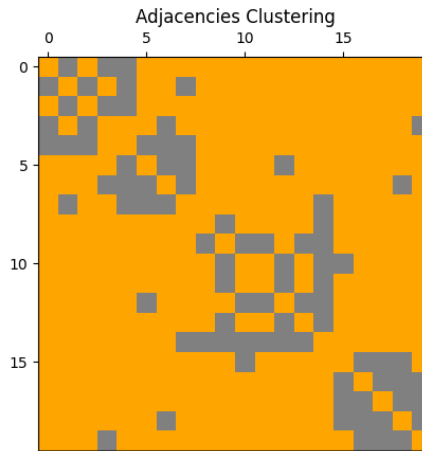


Рис. 6. Матрица смежности графа G_2 .

Граф G_3 установленный из файла имеет 1000 вершин. Матрица смежности A_3 представлена на рисунке 7.

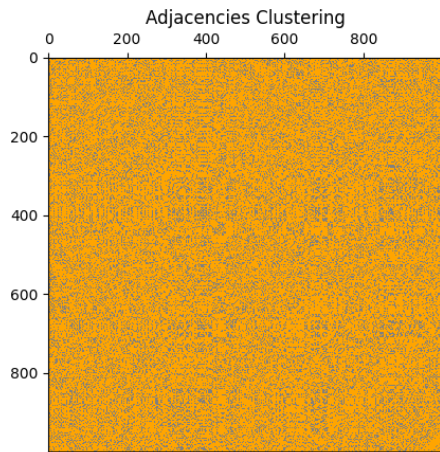


Рис. 7. Матрица смежности графа G_3 .

Функция $Laplas(Matrix)$ (листинг 6), возвращает лапласиан (матрицу Кирхгофа). Она принимает на вход матрицу смежности графа. При помощи формулы (6) реализует построения лапласиана.

Листинг 6. Функция нахождения лапласиана графа.

```

1 def Laplas(Matrix):
2     D = np.zeros((len(Matrix), len(Matrix)))
3     for i in range(len(Matrix)):
4         val = 0
5         for j in range(len(Matrix)):
6             if Matrix[i][j] == 1:
7                 val = val + 1
8         D[i][i] = val
9     return D - Matrix

```

5.2 Доказательство, что невзвешенный граф с n вершинами является положительно полуопределенной матрицей, имеющей n неотрицательных собственных чисел, одно из которых равно нулю.

Доказательство того, что лапласиан является положительно полуопределенной матрицей:

Положительно полуопределенная матрица это матрица, собственные числа которой неотрицательные.

Пусть $Le = 0$, то тогда $e^T Le = 0$. По определению L , для любого вектора x получается:

$$x^T Lx = x^T N N^T x = (N^T x)^T \cdot (N^T x) = \|N^T x\|^2 \geq 0,$$

где N — ориентированная матрица инцидентности, которая принимает значение 0, если i -тая вершина не принадлежит j -тому ребру графа, значение 1, если i -тая вершина является началом j -того ребра, значение -1, если i -тая вершина является концом j -того ребра. $L = NN^T$.

Отсюда получается, что $e^T L e \geq 0$ для всех x , следовательно L положительно полуопределена.

Поскольку L - симметричная матрица и, является положительно полуопределенной, собственные значения L могут быть упорядочены как:

$$0 = \mu_1 \leq \mu_2 \leq \mu_3 \leq \dots \leq \mu_n.$$

Доказательство того, что одно из n собственных чисел лапласиана равно 0:

Собственные числа матрицы A находятся при решении $|L - \lambda E| = 0$, где $L \in R^{k \times k}$ — лапласиан, E — матрица, главная диагональ которой заполнена единицами, а все остальные элементы нулевые.

$$|L - \lambda E| = \begin{vmatrix} k_{11} - \lambda & k_{12} & \dots & k_{1|v|} \\ k_{21} & k_{22} - \lambda & \dots & k_{2|v|} \\ \vdots & \vdots & \ddots & \vdots \\ k_{|v|1} & k_{|v|2} & \dots & k_{|v||v|} \end{vmatrix} = 0$$

К первой строке прибавляются все остальные строки, учитывая, что сумма элементов строки/столбца равна нулю, получается:

$$\begin{vmatrix} k_{11} + k_{21} + k_{|v|1} - \lambda & k_{12} + k_{22} + k_{|v|2} - \lambda & \dots & k_{1|v|} + k_{2|v|} + k_{|v||v|} - \lambda \\ k_{21} & k_{22} - \lambda & \dots & k_{2|v|} \\ \vdots & \vdots & \ddots & \vdots \\ k_{|v|1} & k_{|v|2} & \dots & k_{|v||v|} \end{vmatrix} = 0.$$

По свойству определителя матрицы выносятся λ :

$$\begin{vmatrix} -\lambda & -\lambda & \dots & -\lambda \\ k_{21} & k_{22} - \lambda & \dots & k_{2|v|} \\ \vdots & \vdots & \ddots & \vdots \\ k_{|v|1} & k_{|v|2} & \dots & k_{|v||v|} \end{vmatrix} = 0 \rightarrow$$

$$-\lambda \begin{vmatrix} 1 & 1 & 1 & 1 \\ k_{21} & k_{22} - \lambda & \dots & k_{2|v|} \\ \vdots & \vdots & \ddots & \vdots \\ k_{|v|1} & k_{|v|2} & \dots & k_{|v||v|} \end{vmatrix} = 0.$$

Отсюда очевидно, что 0 является собственным значением.

5.3 Нахождение спектра для каждого графа. Выявления особенностей спектра и связь с количеством кластеров.

Используя лапласианы для графов G_1 , G_2 и G_3 можно найти спектр для каждого графа, так как спектр графа является множеством отсортированных собственных чисел лапласиана графа.

Для графа G_1 были отсортированы собственные числа лапласиана графа L_1 (Рис.8).

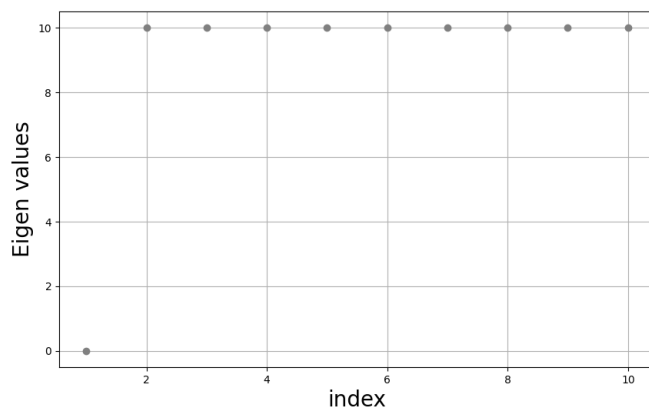


Рис. 8. Спектр графа G_1 .

Для графа G_2 были отсортированы собственные числа лапласиана графа L_2 (Рис.9).

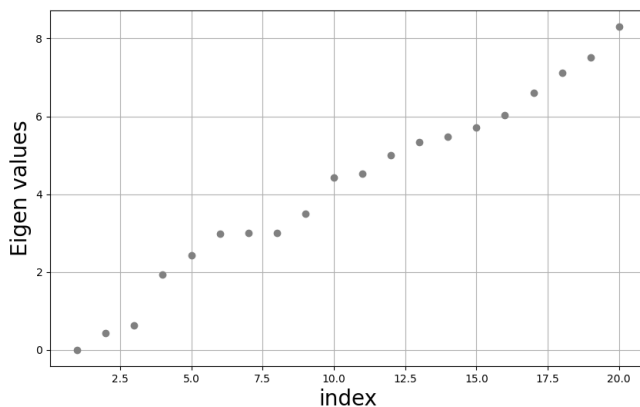


Рис. 9. Спектр графа G_2 .

Для графа G_3 были отсортированы собственные числа лапласиана графа L_3 (Рис.10).

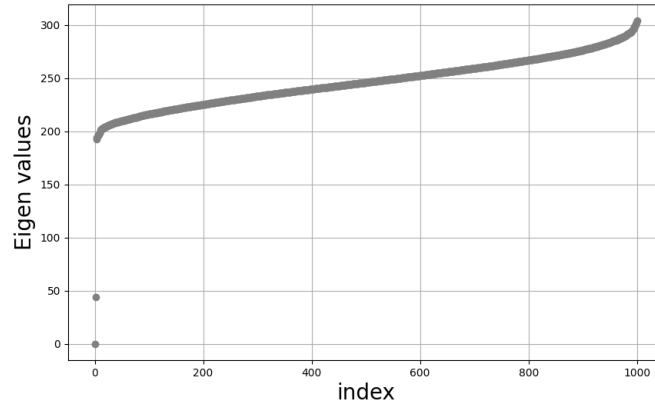


Рис. 10. Спектр графа $G1$.

По графикам 8, 9, 10 видно, что первое отсортированное число на графе лапласиана равно нулю, что подтверждает доказательство предыдущего пункта, так же можно заметить, что все числа не отрицательные.

Для выявления особенностей спектра и связь с количеством кластеров, необходимо было исследовать вектор Фидлера и написать функцию *fiedler*($v, ind, Matrix$), которая поможет в выявлении особенностей. Вектор Фидлера соответствует наименьшему собственному числу отличенному от нуля.

Функция *fiedler*($v, ind, Matrix$) (листинг 7), возвращает отсортированный вектор Фидлера по собственному числу. Она принимает на вход центрированную матрицу, собственное значение и нормализованные собственные векторы.

Листинг 7. Функция для сортировки собственных значений вектора Фидлера.

```

1 def fiedler_vector(w, v, Matrix):
2     f_i = np.argsort(w)[1]
3     if len(Matrix) == 20:
4         f_i = f_i + 1
5     fiedler = []
6     for i in range(len(v)):
7         fiedler.append(v[i][f_i])
8     return fiedler

```

Используя функцию *fiedler*($v, ind, Matrix$), можно выявить связь с количеством кластера.

Для графа G_1 были отсортированы собственные вектора Фидлера (Рис. 11).

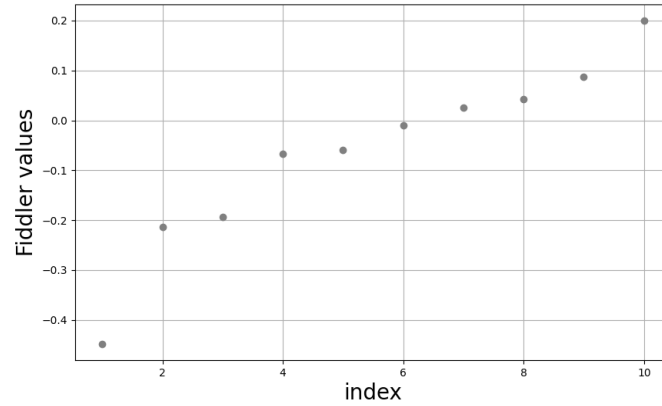


Рис. 11. Визуальный вектор Фидлера для графа G_1 .

Так как граф G_1 является полно связным графом, он не имеет кластеров, что и видно на графике (Рис.11).

Для графа G_2 были отсортированы собственные вектора Фидлера (Рис. 12).

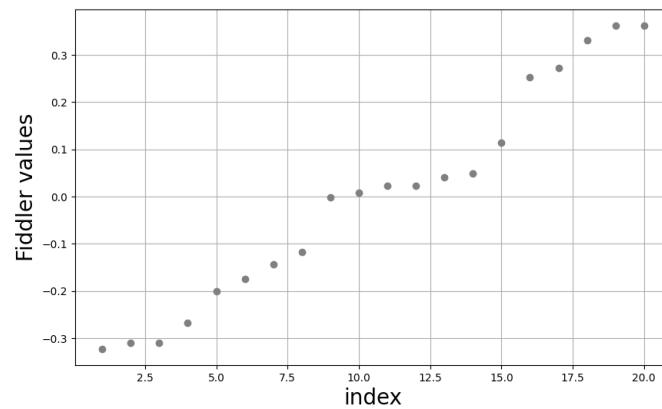


Рис. 12. Визуальный вектор Фидлера для графа G_1 .

Из условия нам известно что граф G_2 имеет 3 кластера (Рис.1). На графике видны эти 3 кластера с вершинами 8, 7 и 5.

Для графа G_3 были отсортированы собственные вектора Фидлера (Рис. 13).

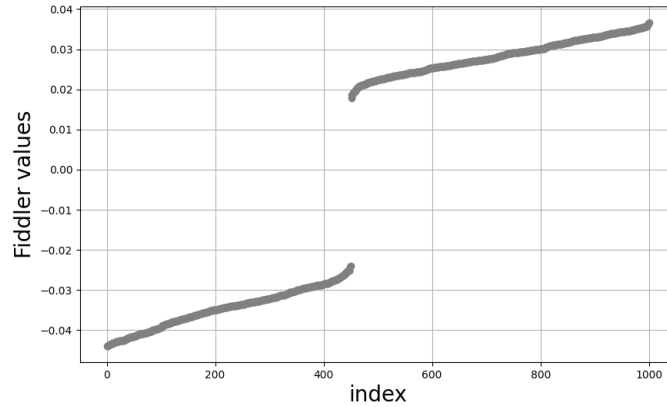


Рис. 13. Визуальный вектор Фидлера для графа G_3 .

На графике (13) для G_3 отчётливо видны два кластера, которые группируются, по значениям вектора Фидлера, на значения отрицательные и положительные. По графику невозможно определить количество вершин.

5.4 Нахождение количество кластеров для третьего графа. Визуализация графа исходной матрицы смежности и её отсортированной версии.

По полученным данным можно сделать вывод, что у графа G_3 два кластера (Рис.13). Для визуализации этих кластеров необходимо в матрице смежности отсортировать строки и столбцы по вектору Фидлера. Эта операция выполняется в функции (*Sort_matrix(v, ind, Matrix)*), которая отсортирует матрицу смежности.

Функция *Sort_matrix(v, ind, Matrix)* (листинг 8), возвращает отсортированную матрицу смежности. Она принимает на вход центрированную матрицу, собственное значение и нормализованные собственные векторы. В функции вычисляет вектор Фидлера, затем выполняется сортировка с сохранением индексов сортировки. При помощи этих индексов сортируются строки и столбцы матрицы смежности.

Листинг 8. Функция сортировки матрицы смежности.

```

1 def Sort_matrix(v, ind, Matrix):
2     for i in range(len(v)):
3         v_stolb.append(v[i][ind])
4     sort = pd.Series(v_stolb).sort_values(ascending=True)
5     ind_s = sort.index.tolist()
6     A = np.zeros((len(Matrix), len(Matrix)))
7     for i in range(0, len(Matrix)):
8         for j in range(0, len(Matrix)):
9             A[i][j] = Matrix[ind_s[i]][ind_s[j]]
10    return v_stolb, A

```

Прогоняя граф G_3 через функцию $Sort_matrix(v, ind, Matrix)$ получается график (Рис.14). Ранее уже был выведен график исходной матрицы графа G_3 (Рис.7). Данный график подтверждает ранее полученные данные, что граф G_3 имеет два кластера.

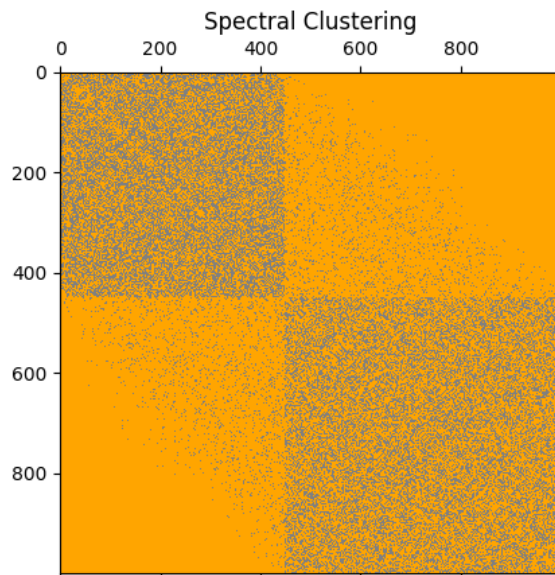


Рис. 14. Кластеризация графа G_3 .

6 Заключение

1. Изучен метод главных компонент для понижения размерности данных, основой которого является сингулярное разложение.
2. При помощи метода главных компонент проведено сепарирование набора данных на предмет опухоли молочной железы.
3. Из набора данных опытным путем было выявлено, что первые два компонента достаточны для сепарации данных.
4. Рассмотрена спектральная кластеризация трех графов G_1, G_2, G_3 с различными вершинами.
5. Опытным путём были отсортированы вектора Фидлера для каждого графа и выявлены особенности связи с количеством кластеров.
6. Проведена проверка исследования, путем сортировки исходной матрицы смежности.

Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021. С. 140.
2. Документация Python 3.9 [Электронный ресурс] [Официальный сайт]. 2021. (дата обращения 27.10.2021).
3. Graph Theory Cesar O. Aguilar [Электронный ресурс] [Официальный сайт]. 2021. (дата обращения 19.12.2021).

Выходные данные

Караф С. М.. Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2021. — 19 с. URL: <https://sa2systems.ru:88> (система контроля версий кафедры РК6)

Постановка: © ассистент кафедры РК-6, PhD А.Ю. Першин
Решение и верстка: © студент группы РК6-52Б, Караф С. М.

2021, осенний семестр