Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего 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	Тетерин Никита Евгеньевич
Группа	РК6-54б
Тип задания	лабораторная работа №4
Тема лабораторной работы	Спектральное и сингулярное разложения (вариант 5)

Студент _____ **Тетерин Н.Е.**
подпись, дата *фамилия, и.о.*

Преподаватель _____ **Соколов А.П.**
подпись, дата *фамилия, и.о.*

Оценка _____

Москва, 2021 г.

Оглавление

Задание на лабораторную работу

В базовой части данной работы требуется реализовать алгоритм метода главных компонент (англ. Principal component analysis, сокр. PCA) и применить его для понижения размерности данных с последующим применением алгоритмов машинного обучения для выполнения задачи классификации. В продвинутой части необходимо рассмотреть задачу спектрального анализа графов, а именно кластеризацию на графе.

Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – проанализировать теоретические основы PCA, а также применить данный алгоритм и проанализировать его результат. Повторить основы работы с графами, рассмотреть основные свойства такой важной характеристики графа, как его Лапласиан. Произвести спектральное разложение нескольких графов и анализ результатов.

Выполненные задачи

1. Получить данные и нормализовать их.
2. Реализовать алгоритм PCA, применить его для понижения размерности данных.
3. Сравнить результат с библиотечной реализацией, получить метрики качества классификации.
4. Построить матрицы смежности и Лапласианы приведенных графов.
5. Показать, что ненормализованный Лапласиан неориентированного связного графа является положительно полуопределенной матрицей.
6. Произвести спектральное разложение приведенных графов, визуально определить кластеры.
7. Произвести кластеризацию проекции графа на трехмерное пространство с помощью алгоритма DBSCAN.

1. Идея метода главных компонент

Представим, что имеется некоторый набор данных большой размерности. Обыкновенно наличие большого числа признаков благоприятно влияет на качество работы моделей машинного обучения, поскольку чем больше признаков, тем больше потенциально полезной уточняющей информации. Однако в определенный момент при увеличении размерности данных метрики качества начинают падать (этот феномен зачастую называют “проклятием размерности”, англ. “The curse of Dimensionality”). Это происходит по той причине, что при увеличении размерности задачи плотность данных падает экспоненциально при неизменном числе наблюдений. Поэтому очень часто возникает необходимость выявить такой базис пониженной размерности в данных, который позволяет сохранить большую часть информации об их исходном распределении. Одним из способов решения данной проблемы является понижение размерности. Метод главных компонент является, пожалуй, самым распространенным алгоритмом для решения задачи понижения размерности. Он предполагает построение ортогонального базиса пониженной размерности, содержащего максимально возможную информацию о данных большой размерности. Чтобы понять, как это происходит, нужно разобраться с такими понятиями, как сингулярные числа и вектора.

Сингулярные числа и сингулярные вектора – некоторое обобщение понятия собственных чисел и векторов соответственно на прямоугольные матрицы. В свою очередь, собственным числом λ_i и соответствующим ему ненулевым собственным вектором a_i матрицы $A \in \mathbb{R}^{n\times n}$ называются такие число и вектор, что справедливо (1).

$$(1) \quad Aa_i = \lambda a_i$$

Важным свойством собственных векторов является тот факт, что в случае их линейной независимости (а, следовательно, линейности оператора A) они формируют базис пространства $\mathbb{R}^n$. При этом каждый из собственных векторов будет определять направление, в котором действие линейного оператора A будет приводить лишь к масштабированию произвольного вектора с коэффициентом масштабирования, равным соответствующему собственному числу λ. Сингулярные числа вводятся как $\sigma_i = \sqrt{\lambda_i}$, где λ_i это соответственно собственное число и т. н. матрицы Грама $K = A^T A$. Соответствующие им собственные вектора являются сингулярными векторами матрицы A. Матрица Грама при этом является положительно (неотрицательно) определенной матрицей с собственными числами, которые можно упорядочить в следующую последовательность:

$$(2) \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$$

При этом число ненулевых сингулярных векторов является рангом исходной матрицы A, то есть числом её линейно-независимых строк/столбцов, а также рангом матрицы K. В методе главных компонент необходимо получить матрицу ковариации признаков, при этом все признаки важно предварительно нормализовать (поскольку стандартные отклонения и матожидания признаков различаются). Нормализация предполагает центрирование и масштабирование. Под центрированием понимают сдвиг распределения данных таким образом, чтобы матожидание полученного распределение равнялось 0, а под масштабированием – преобразование данных таким образом, чтобы стандартное отклонение полученного распределения равнялось 1. Важно отметить, что масштабирование стандартного отклонения не предполагает, что полученные данные будут лежать в некотором фиксированном диапазоне (например, от -1 до 1). Описанное преобразование обычно называется min-max scaling, т. е. сжатие всего исходного диапазона распределения данных в фиксированный. Если предполагать, что данные распределены

нормально, то можно считать, что практически все они (99.73%) будут находиться в т. н. диапазоне 6 сигм (6 sigma rule).
В терминах линейной алгебры для прямоугольной матрицы наблюдений $X \in \mathbb{R}^{m \times n}$ описанные преобразования можно записать следующим образом:

$$\text{вектор средних } \hat{x} = \frac{1}{m} e^T X, \qquad e - \text{единичный вектор;}$$

$$(3) \quad A = (X - e\hat{x})\Sigma = \left(E - \frac{1}{m} ee^T\right) X\Sigma$$

здесь Σ – диагональная матрица, i-й элемент которой равен $1/\sigma_i$, величине, обратной стандартному отклонению факта i. Таким образом, матрица $A \in \mathbb{R}^{m \times n}$ является матрицей нормализованных данных.
Покажем, что дисперсия проекции данных на собственный вектор матрицы Грама K, соответствующий максимальному собственному числу (на сингулярный вектор матрицы A, соответствующий максимальному сингулярному числу) максимальна: по условию требуется определить такой вектор $\vec{u}$, что дисперсия проекции матрицы A на него максимальна, иными словами, выполняется (4), при этом $\nu = 1/(m-1)$, усредняющий множитель в выражении для дисперсии. Подобная краткая матричная запись объясняется тем фактом, что L_2 норма вектора $\vec{p}$ задает сумму квадратов отклонений своих компонент от матожидания, равного 0, из которой легко получить выражение для среднеквадратичного отклонения, а значит и для дисперсии.

$$(4) \quad \vec{p} = A\vec{u} \Rightarrow \sigma_p^2 = \nu\|\vec{p}\|_2^2 = \nu\|A\vec{u}\|_2^2 = \nu(A\vec{u})^T(A\vec{u}) = \nu\vec{u}^T A^T A\vec{u} = \nu\vec{u}^T K\vec{u}, \qquad \max_{\vec{u},\ \|\vec{u}\|=1} (\nu\vec{u}^T K\vec{u})$$

По теореме о собственных числах и векторах симметричной матрицы имеем решение данной задачи максимизации:

$$\nu\lambda_1 = \max_{\vec{u},\ \|\vec{u}\|=1} (\nu\vec{u}^T K\vec{u}), \qquad \vec{u} - \text{собственный вектор матрицы } K, \text{соотв. } \lambda_1$$

Получается, что необходимо сформировать матрицу ковариации нормированных данных (которая, в сущности, является матрицей корреляции, поскольку корреляция двух величин есть ковариация, деленная на стандартное отклонение каждой из величин), определить её собственные числа и векторы, отсортировать собственные вектора таким образом, чтобы порядок их соответствовал порядку отсортированных (по не возрастанию) собственных чисел.
Далее можно рекурсивно применять данный подход для определения всех главных компонент. В общем случае главные компоненты формируют ортонормированный базис в пространстве исходной размерности такой, что доля объясненной дисперсии вдоль каждой следующей главной компоненты не возрастает. Под объясненной дисперсией понимается отношение дисперсии вдоль одной из главных компонент и суммарной дисперсии вдоль всех компонент (суммы собственных чисел матрицы корреляции). Можно воспринимать проделанную операцию следующим образом: проекция матрицы A на базис главных компонент производит некоторое линейное преобразование в пространстве, из которого были взяты исходные данные. Вообще говоря, размерность этого пространстве не обязательно равна размерности исходных данных: она равняется $rg(A)$. Чтобы добиться понижения размерности, необходимо

спроецировать матрицу A на базис, состоящий из требуемого числа главных компонент. В следующем разделе повторяются все описанные выше действия для конкретных данных.

2. Реализация PCA

Исходный датасет состоит из 30 численных признаков, индекса пациента и метки класса (присутствие рака молочной железы у пациента). Небольшая часть данных приведена в табл. 2.1. Невооруженным взглядом можно наблюдать различные масштабы признаков. Более конкретные статистические данные приведены в табл. 2.2.

index	ID	Diag	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14
0	842302	M	17.99	10.38	122.8	1001.0	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589	153.4
1	842517	M	20.57	17.77	132.9	1326.0	0.08474	0.07864	0.0869	0.07017000000000001	0.1812	0.05667000000000005	0.5435	0.7339	3.398	74.08
2	84300903	M	19.69	21.25	130.0	1203.0	0.1096	0.1599	0.1974	0.1279	0.2069	0.05998999999999995	0.7456	0.7869	4.585	94.03
3	84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445	27.23
4	84358402	M	20.29	14.34	135.1	1297.0	0.1003	0.1328	0.198	0.1043	0.1809	0.05882999999999999	0.7572	0.7813	5.438	94.44
5	843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07612999999999999	0.3345	0.8902	2.217	27.19
6	844359	M	18.25	19.98	119.6	1040.0	0.09462999999999999	0.109	0.1127	0.07400000000000001	0.1794	0.05742000000000006	0.4467	0.7732	3.18	53.91
7	84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366000000000001	0.05985	0.2196	0.07451	0.5835	1.3769999999999998	3.8560000000000003	50.96
8	844981	M	13.0	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406	24.32
9	84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08542999999999999	0.203	0.08242999999999999	0.2976	1.599	2.039	23.94
10	845636	M	16.02	23.24	102.7	797.8	0.08206000000000001	0.06669	0.03299	0.03323	0.1528	0.05697000000000001	0.3795	1.187	2.4659999999999997	40.51
11	84610002	M	15.78	17.89	103.6	781.0	0.0971	0.1292	0.09954	0.06606000000000001	0.1842	0.06082000000000006	0.5058	0.9849	3.5639999999999996	54.16
12	846226	M	19.17	24.8	132.4	1123.0	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07	116.2
13	846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903	36.58
14	84667401	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061	19.21
15	84799002	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.8789999999999996	32.55
16	848406	M	14.68	20.13	94.74	684.5	0.09867000000000001	0.07200000000000001	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195	45.4
17	84862001	M	16.13	20.68	108.1	798.8	0.11699999999999999	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.8539999999999996	54.18
18	849014	M	19.81	22.15	130.0	1260.0	0.09831000000000001	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.0170000000000001	5.865	112.4
19	8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06663999999999999	0.04781000000000005	0.1885	0.05766	0.2699	0.7886	2.0580000000000003	23.56
20	8510653	B	13.08	15.71	85.63	520.0	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383	14.67
21	8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.029560000000000003	0.02076	0.1815	0.06905	0.2773	0.9768	1.909	15.7
22	8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756000000000001	0.2521	0.07032000000000001	0.4388	0.7096	3.384	44.91
23	851509	M	21.16	23.04	137.2	1404.0	0.09427999999999999	0.1022	0.1097	0.08632000000000001	0.1769	0.05277999999999994	0.6917	1.127	4.303	93.99

Табл. 2.1. Элемент таблицы с исходными данными. Можно наблюдать различный масштаб признаков.

Код для данной ЛР традиционно написан на ЯП *python*, и в нем активно используются особенности пакета *numpy*, в частности понятие *broadcasting*, позволяющее применять арифметические операции с многомерными данными разных размерностей. Например, на листинге 2.1 приведен код функции для получения матрицы корреляции для ненормализованных данных, использующей данное свойство. Так, при вычитании вектора средних значений из матрицы производится расширение размерности первого таким образом, что в результате вектор средних вычитается из каждой строки матрицы. Аналогично, можно лаконично отмасштабировать данные путем деления на стандартное отклонение вдоль каждого столбца. Далее матрицы перемножаются так, чтобы получилась матрица Грама и дополнительно умножаются на коэффициент $\nu = 1/(n-1)$ (коэффициент для несмещенных выборочных статистик: матожидания, дисперсии). Интересно, что понятие «ковариация», вообще говоря, дословно может быть переведено как «кодисперсия» (англ. *covariance* и *variance* соответственно). Функция $pca(A)$ получает на вход матрицу исходных данных, осуществляет нормализацию и возвращает кортеж, первый элемент которого содержит главные компоненты, а второй – соответствующие им стандартные отклонения. На рис. 2.3 приведена тепловая карта полученной матрицы корреляции. Видно, что вся главная диагональ заполнена единицами, что неудивительно, ведь корреляция величины с самой собой буквально является отношением её дисперсии к квадрату её стандартного отклонения, то есть той же дисперсии.

	count	mean	std	min	50%	max
f1	569	14.13	3.52	6.981000	13.370000	28.11
f2	569	19.29	4.30	9.710000	18.840000	39.28
f3	569	91.97	24.30	43.790000	86.240000	188.50
f4	569	654.89	351.91	143.500000	551.100000	2501.00
f5	569	0.10	0.01	0.052630	0.095870	0.16
f6	569	0.10	0.05	0.019380	0.092630	0.35
f7	569	0.09	0.08	0.000000	0.061540	0.43
f8	569	0.05	0.04	0.000000	0.033500	0.20
f9	569	0.18	0.03	0.106000	0.179200	0.30
f10	569	0.06	0.01	0.049960	0.061540	0.10
f11	569	0.41	0.28	0.111500	0.324200	2.87
f12	569	1.22	0.55	0.360200	1.108000	4.88
f13	569	2.87	2.02	0.757000	2.287000	21.98
f14	569	40.34	45.49	6.802000	24.530000	542.20
f15	569	0.01	0.00	0.001713	0.006380	0.03
f16	569	0.03	0.02	0.002252	0.020450	0.14
f17	569	0.03	0.03	0.000000	0.025890	0.40
f18	569	0.01	0.01	0.000000	0.010930	0.05
f19	569	0.02	0.01	0.007882	0.018730	0.08

Табл. 2.2. Основные статистики исходных данных (приведены для признаков 1–19).

```python
# this function yields correlation matrix for the given data
# one can observe that this is the very thing corr() method
# does for a DataFrame
# also, the np.cov() method does the same thing
# but only when applied to a normalized matrix,
# of course
def make_corr_matrix(M: np.atleast_2d):
  std = M.std(axis=0)
  means = M.mean(axis=0)
  cm = ((M - means) / std).T @ ((M - means) / std) / (M.shape[0] - 1)
  return cm
```

Листинг 2.1. Функция для получения матрицы корреляции.

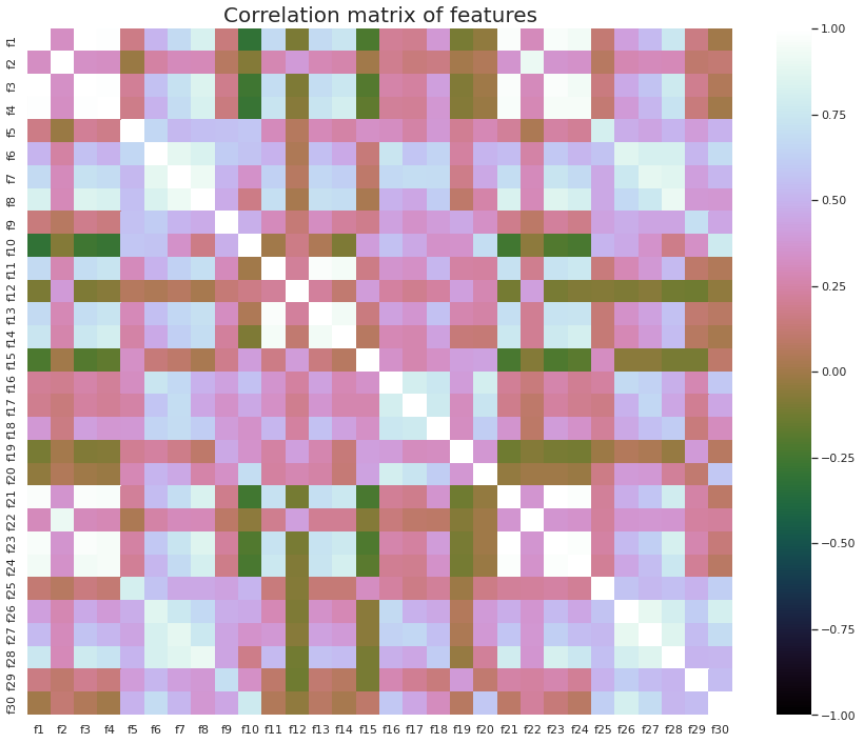

Рис. 2.3. Тепловая карта матрицы корреляции данных.

```python
def pca(A: np.ndarray) -> tuple:
    cm = make_corr_matrix(A)

    eig_val, eig_vec = np.linalg.eig(cm)
    eig_val = np.flip(eig_val.argsort())

    pc = eig_vec[:, eig_val]
    std = np.sqrt(eig_val)
    return pc, std
```

Листинг 2.2. Функция для нахождения базиса главных компонент и соответствующих стандартных отклонений.

На рис. 2.4 приведены значения стандартных отклонений вдоль каждой из главных компонент. После получения базиса можно выбрать из него k главных компонент и совершить отображение вида $\mathbb{R}^{rg(A)} \rightarrow \mathbb{R}^k$. На рис. 2.5 и 2.6 приведены визуализации этого метода. Цвета при этом соответствуют двум классам, которые необходимо предсказывать. Можно наблюдать, что пусть данные в полученном пространстве и не являются линейно разделимыми, но метрики качества классических алгоритмов машинного обучения (была использованы логистическая регрессия и метод опорных векторов из библиотеки *scikit-learn*) довольно высокие: f1-score составил более 93%! Отдельно хочется отметить тот факт, что метрики качества на двумерных данных несколько выше, чем на трехмерных, однако в обоих случаях модели показали себя почти вдвое лучше, чем на исходных данных. Результаты классификации находятся в Jupiter-ноутбуке вместе со остальным исходным кодом. Для самопроверки был применен алгоритм PCA из библиотеки *sklearn*, и результаты совпали с таковыми, полученными самостоятельно, из чего, пожалуй, можно сделать вывод о правильности собственной реализации.

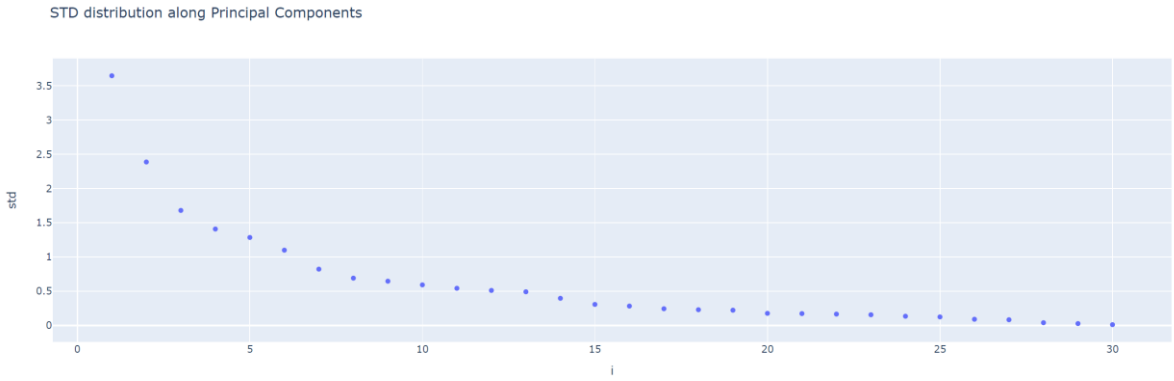

Рис. 2.4. Значения стандартных отклонений вдоль главных компонент. Ось абсцисс: номер главной компоненты; ось ординат: стандартное отклонение вдоль этой оси.

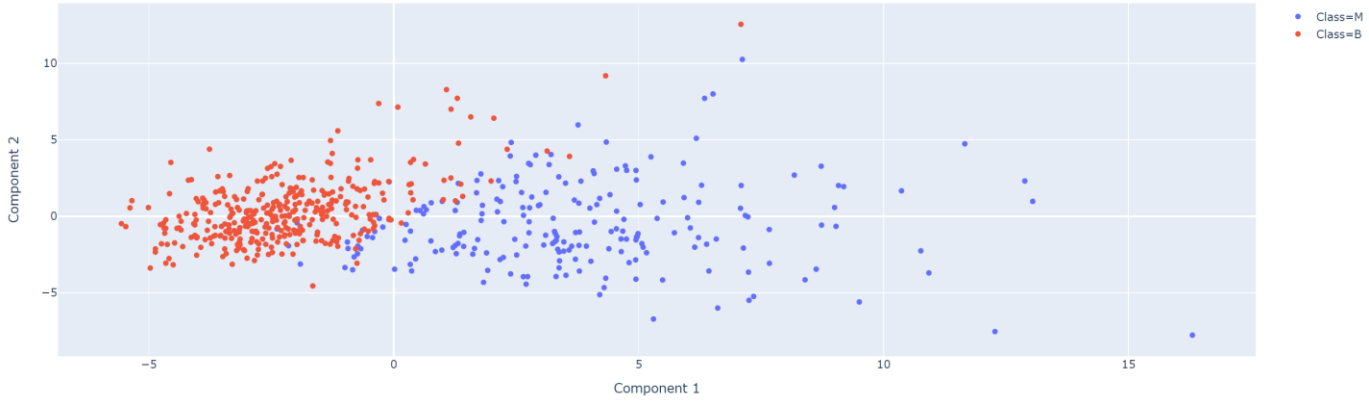

Рис. 2.5. Проекция данных на первые 2 главные компоненты. Можно наблюдать, что классы слабо проникают друг в друга.

Рис. 2.6. Проекция данных на первые 3 главные компоненты. Данные нормализованы, интересное наблюдение заключается в том, что диапазон оси очередной компоненты меньше, чем у предыдущей, что соотносится с меньшей величиной дисперсии.

3. Матрица смежности и Лапласиан графа

Граф – это понятие из области дискретной математики, хрестоматийно определяемое как набор двух (обычно конечных) множеств: множества вершин $V(G)$ и множества (мультимножества в случае мультиграфа) ребер $E(G)$, каждое из которых соединяет две вершины. Мы будем рассматривать т. н. неориентированные невзвешенные графы (в общем случае каждое ребро может иметь ряд атрибутов, таких как направление и вес).

Граф сам по себе не имеет понятия размерности и единого изображения (различные изображения одного и того же графа, например, на плоскости, называются *изоморфизмами* графа, т. е. такими представлениями, между которыми можно построить *биекцию*, однозначное соответствие между вершинами и ребрами обоих графов). *Матрицей смежности* графа $G = (V(G), E(G))$ называется такая матрица $v \times v$ (здесь v – мощность множества вершин), элементы которой задаются по правилу (5) в случае неориентированного графа (в случае ориентированного графа будут возникать значения -1 и 1, обозначающие выходящие/входящие ребра).

$$(5) \quad a_{ij} = \begin{cases} 1 \to \{i, j\} \in E(G) \\ 0 \text{ иначе} \end{cases}$$

В случае неориентированного графа данная матрица является симметричной. *Лапласианом* графа L называется матрица вида $D - A$, где A – матрица смежности (англ. *Adjacency matrix*), D – диагональная матрица, состоящая из степеней вершин (степенью $deg(i)$ вершины $i \in V(G)$ называется число инцидентных ей ребер). Построим Лапласианы и матрицы смежности нескольких графов, приведенных в качестве примера в условии данной работы.

Первый граф является кликой на 10 вершинах (рис. 3.2), его характеристические матрицы изображены на рис. 3.1.

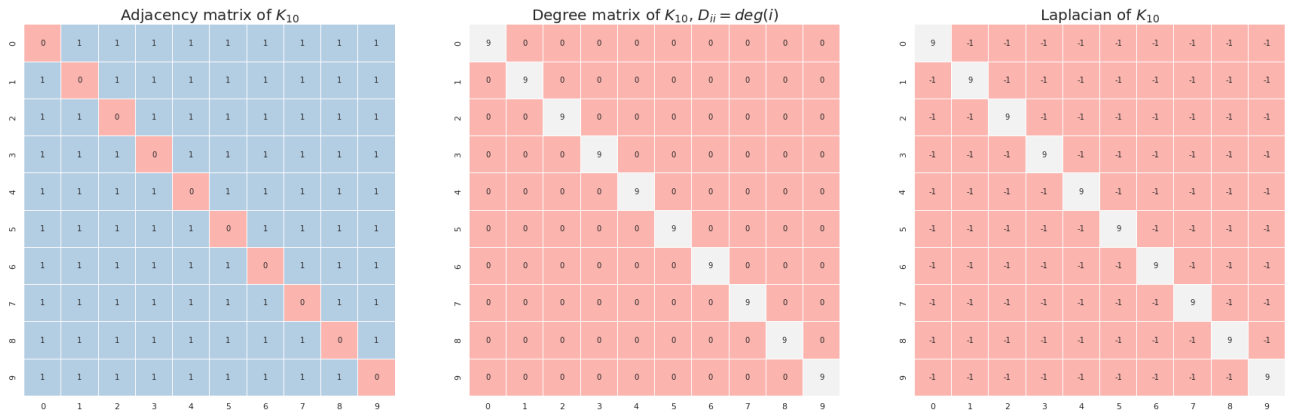

Рис. 3.1. Матрица смежности, диагональная матрица степеней вершин и Лапласиан полного графа на 10 вершинах.

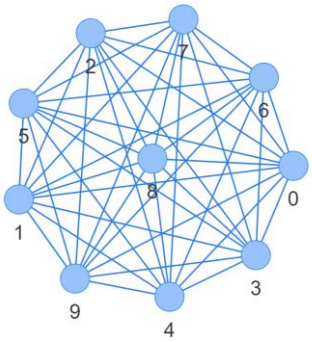

Рис. 3.2. Один из вариантов представления K_{10}.

Изображение второго графа приведено на рис. 3.3, матрица смежности была получена мануально и приведена на рис. 3.4, как и Лапласиан данного графа.

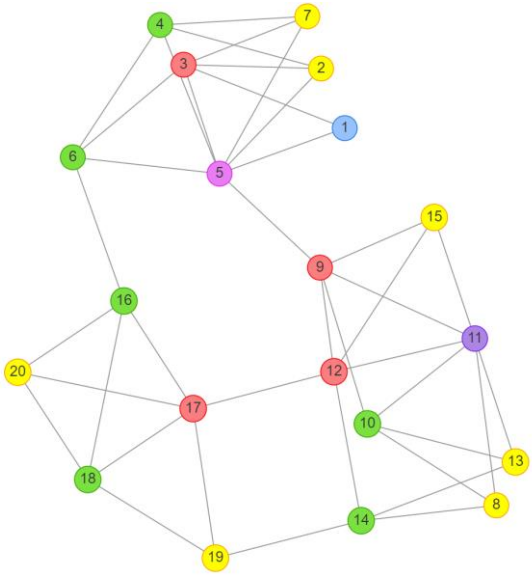

Рис. 3.3. Один из вариантов представления графа G_2,
полученный с помощью библиотеки для визуализации графов pyvis.
Различными цветами обозначены вершины различных степеней.

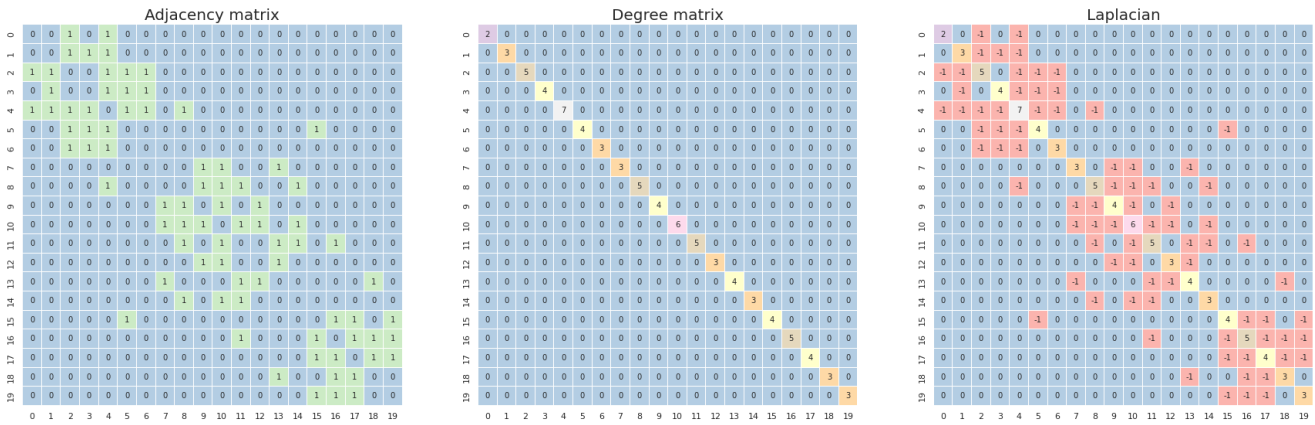

Рис. 3.4. Матрица смежности (слева), матрица D (в центре) и Лапласиан (справа) графа G_2.

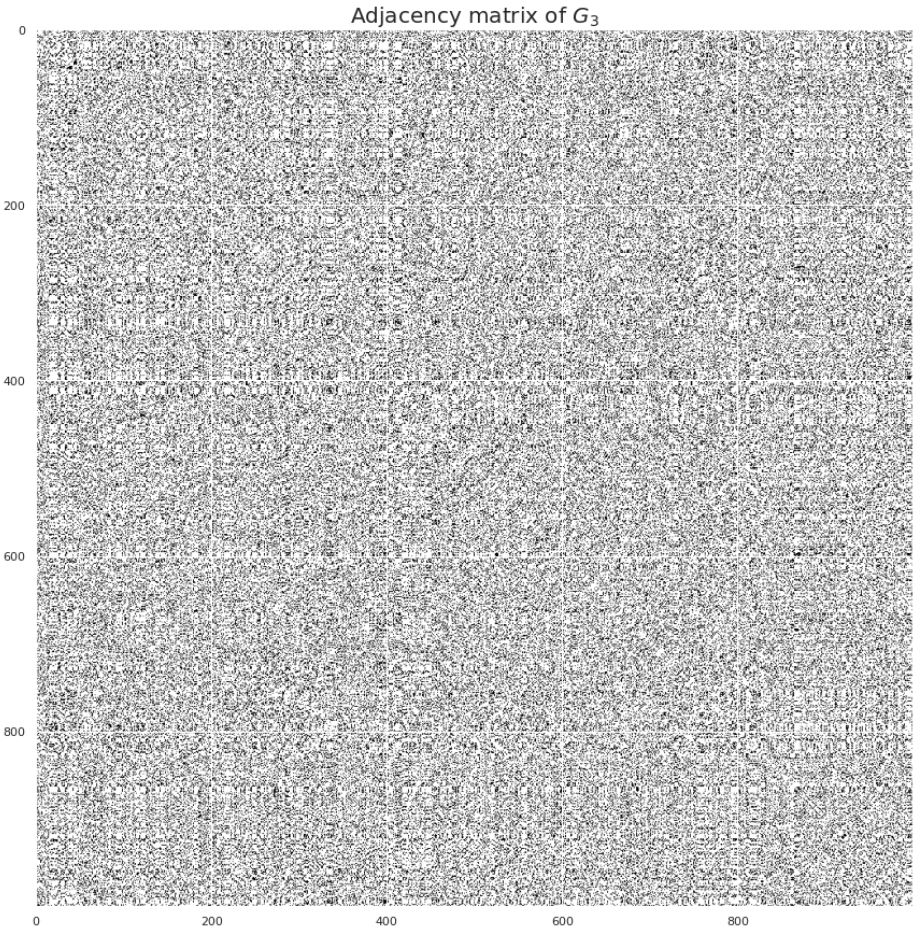

Рис. 3.5. Матрица смежности графа G_3.

Матрицу смежности графа G_3, приведенную на рис. 3.5, весьма трудно интерпретировать в исходном виде. Гистограмма степеней вершин этого графа приведена на рис. 3.6, можно наблюдать, что степени вершин распределены в пределах от 196 до 300, что делает каждую вершину связанной в среднем с 1/4 вершин всего графа. Далее мы рассмотрим, как, используя спектральное разложение и переставляя столбцы и строки определенным образом можно получить вполне интерпретируемую матрицу смежности.

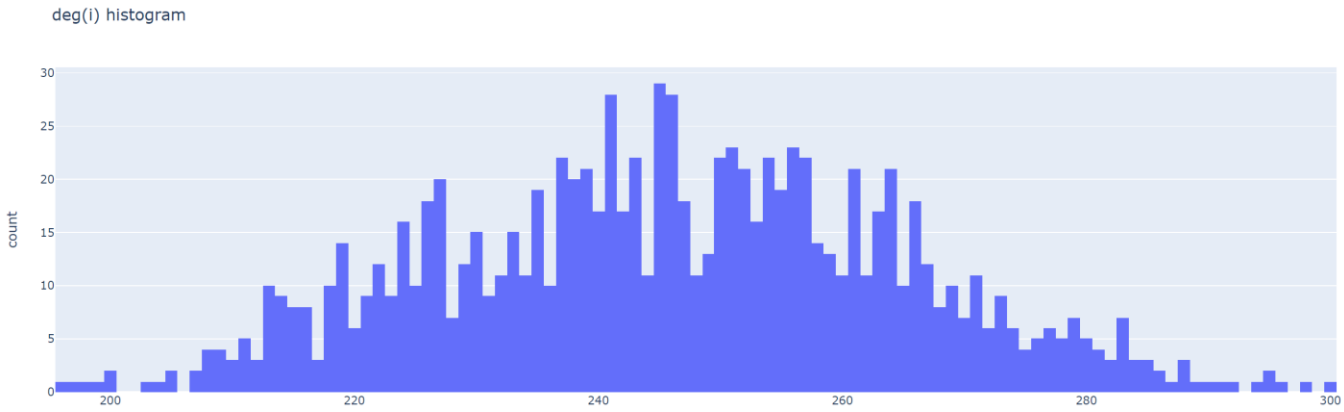

Рис. 3.6. Гистограмма степеней вершин графа G_3.

Покажем, что классический Лапласиан неориентированного связного невзвешенного графа является положительно полуопределенной матрицей. Для этого введем матрицу $Q(G)$, $Q \in \mathbb{Z}^{v \times e}$ (e – мощность множества ребер) такую, что истинно (6). Это матрица инцидентности графа G, если бы он был произвольно ориентирован. В дальнейших рассуждениях порядок, в котором стоят -1 и 1 не имеет значения. Попробуем получить матрицу Грама для Q.

$$(6) \quad Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1e} \\ \vdots & \vdots & \vdots & \vdots \\ q_{v1} & q_{v2} & \cdots & q_{ve} \end{bmatrix}, \qquad q_{ij} = \begin{cases} -1, \text{если есть ребро из } i \text{ в } j \\ 1, \text{если есть ребро из } j \text{ в } i \\ 0 \text{ иначе} \end{cases}$$

Известно, что если матрица A является положительно определенной, то её можно представить в виде $A = B^T B$. Попробуем записать, чему равняется $Q^T Q$:

$$QQ^T = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1e} \\ \vdots & \vdots & \vdots & \vdots \\ q_{v1} & q_{v2} & \cdots & q_{ve} \end{bmatrix} \begin{bmatrix} q_{11} & \cdots & q_{v1} \\ q_{12} & \cdots & q_{v1} \\ \vdots & \cdots & \vdots \\ q_{1e} & \cdots & q_{ve} \end{bmatrix} = M$$

При этом диагональные элементы m_{ii} матрицы $M \in \mathbb{Z}^{v \times v}$ примут вид $\sum_{k=1}^{e} q_{ik}^2 = \deg(i)$, а остальные можно записать в общем виде как $m_{ij} = \sum_{k=1}^{e} (q_{ik} q_{jk})$, т. е.

$$m_{ij} = \begin{cases} -1, \text{если вершины } i \text{ и } j \text{ смежные} \\ 0 \text{ иначе} \end{cases}$$

Именно такой вид имеет Лапласиан L данного графа по определению, иначе говоря, $M = L$, а L является положительно определенной матрицей. Теперь покажем, что Лапласиан имеет нулевое собственное число: очевидно, что сумма элементов в столбцах и строках L равняется 0, таким образом, единичный вектор является его собственным вектором, а 0 – соответствующим ему собственным числом. К сожалению, строго доказать тот факт, что кратность нулевого собственного числа в случае связного графа равна 1, не удалось. Однако, принимая данный факт без доказательства, несложно показать, что у Лапласиана графа, имеющего n компонент связности нулевое собственное число имеет кратность n. Действительно, строки и столбцы данного графа можно переставить таким образом, что матрица примет вид, при котором вдоль главной диагонали будут расположены Лапласианы каждой из компонент смежности, при этом, очевидно, элементы, стоящие на пересечении строк и столбцов, соответствующих вершинам из различных компонент связности, будут нулевыми и не будут влиять на, например, прямой ход метода Гаусса, примененного к общей матрице. При этом каждая подматрица, содержащая Лапласиан компоненты связности, будет иметь один нулевой опорный элемент. Известно, что знаки собственных чисел матрицы совпадают со знаками опорных элементов. В результате получаем n нулевых опорных элементов и, как следствие, n нулевых собственных чисел.

Интересными свойствами обладает наименьшее ненулевое собственное число λ_2 и соответствующий ему вектор, называемый вектором Фидлера. Само собственное число называется алгебраическим числом связности графа. Вектор Фидлера, по сути, задает некоторую функцию на графе, так, что вершине i ставится в соответствие i-я компонента данного вектора. Можно многое рассказать об этом разбиении, но основным и нужным нам фактом является то, оно позволяет получить отобразить граф в пространство $\mathbb{R}^1$, проекция Лапласиана на данный вектор позволяет выделить 2 (иногда 3) подграфа максимального

размера, также называемых кластерами. Чтобы определить большее число кластеров, необходимо проецировать Лапласиан на базис, составленный из большего числа собственных векторов, соответствующих k наименьшим ненулевым собственным числам (именно это и производится далее) с дальнейшим применением алгоритмов кластеризации (DBSCAN, k-means).

4. Применение спектрального разложения графа для определения кластеров

Произведем отображение приведенных выше графов на 1, 2 и 3-мерные пространства, а также приведем список отсортированных по не возрастанию собственных чисел Лапласиана. На рис. 4.1-4 приведены результаты для графа K_{10}, на рис. 4.5-4.9 соответственно для графа G_2, на рис. 4.10-4.14 для графа G_3. Приведем некоторые наблюдения: спектр клики на 10 вершинах, как и любой другой клики, имеет специальный вид (также это касается графов-колец, графов-звезд и графов-путей), а именно $\{0, n, \dots, n\}$, где кратность собственного числа n (числа вершин графа) равняется соответственно $n - 1$. Проекция графа на вектор Фидлера не дала адекватного разбиения на кластеры, поскольку, ожидаемо, разбить K_{10} на 2 кластера примерно одинакового размера не получается. Похожий результат получается при проекции на 2 и 3-мерные пространства. Можно предположить, что для достижения такого разбиения потребуется сделать проекцию на 9-мерное пространство. Спектральное разложение графа G_2 уже оказалось более интерпретируемым, проекции на 1, 2, 3-мерные пространства позволяют визуально различить кластеры, наблюдаемые на рис. 3.3. Матрица смежности данного графа составлялась на основе изображения таким образом, что нумерация вершин идет в порядке кластеров. Поэтому сразу заметно, что, например, первые 7 вершин относятся к одному кластеру, при этом 5 и 6 вершины имеют ближайшие к 0 компоненты. Это можно интерпретировать так, что 5 и 6 вершины наиболее близки к «центру» графа, в то время как остальные (из числа 1–7) находятся больше «на периферии». Это действительно так, ведь именно 5 и 6 вершины соединяют данный кластер с остальными! Аналогичные наблюдения можно получить для вершин 9 и 16. Наконец, граф G_3, содержащий 1000 вершин, оказывается, содержит 2 больших кластера. Причем, как можно наблюдать по матрице смежности, отсортированной по возрастанию компонент вектора Фидлера, один из них более плотный и малочисленный, чем второй. Это наблюдение подтверждается двух- и трехмерными отображениями графа. По компонентам вектора Фидлера можно наблюдать, что все вершины почти равномерно распределены по 2 кластерам, помеченным соответственно положительными и отрицательными компонентами этого вектора.

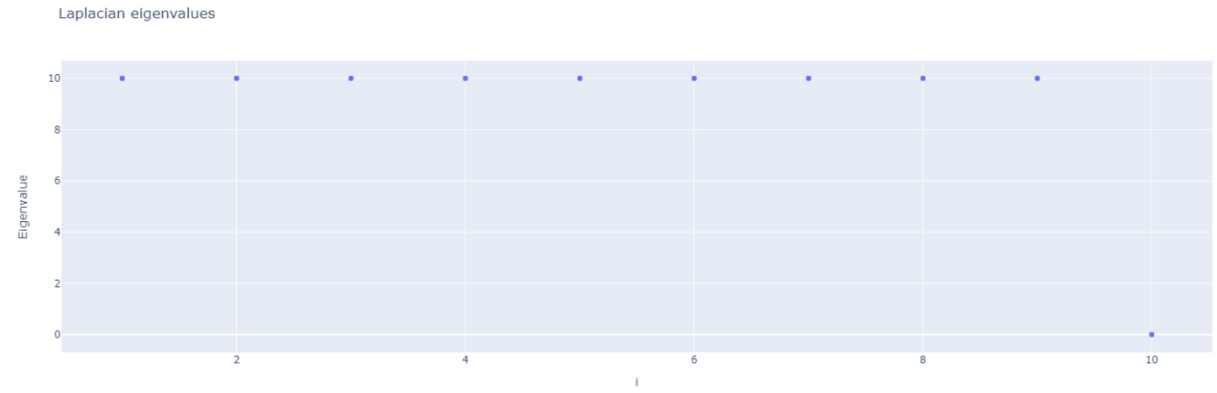

Рис. 4.1. Собственные числа Лапласиана графа K_{10}. Можно наблюдать кратность 9 числа, соответствующего алгебраической связности графа.

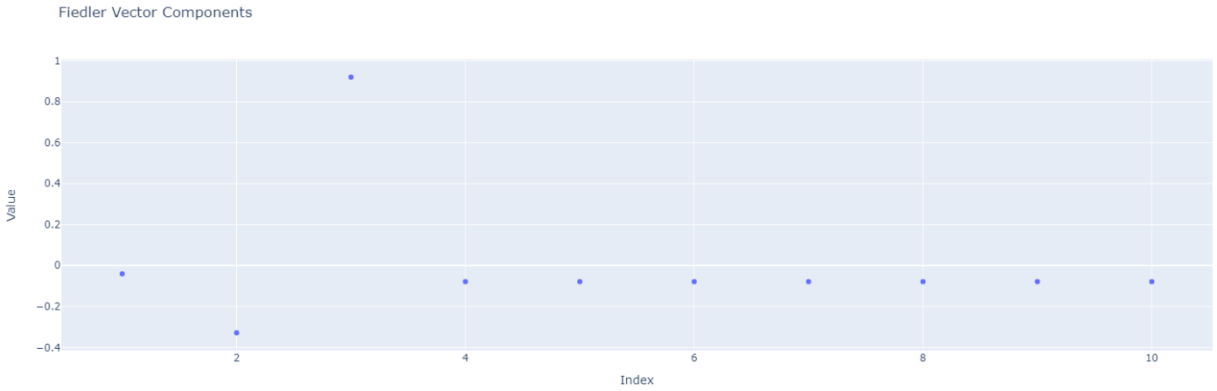

Рис. 4.2. Значения компонент вектора Фидлера для графа K_{10}.

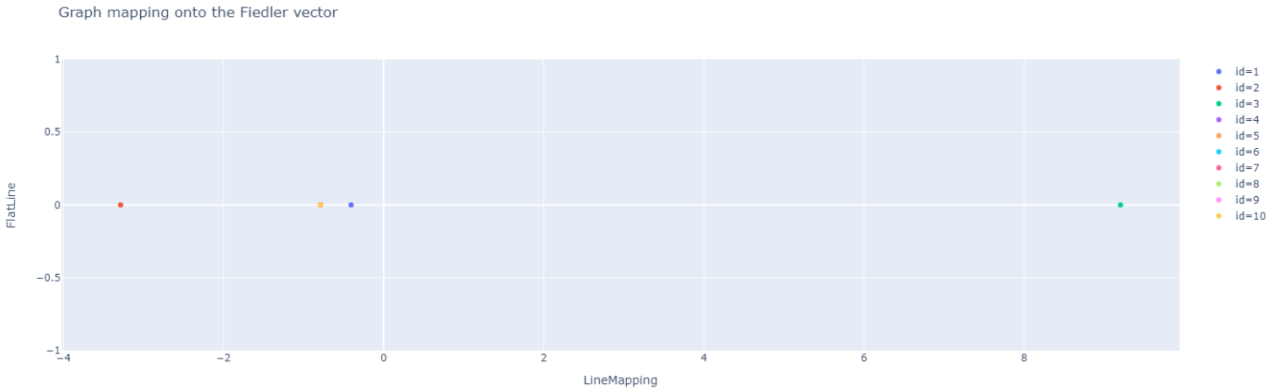

Рис. 4.3. Проекция графа K_{10} на вектор Фидлера.

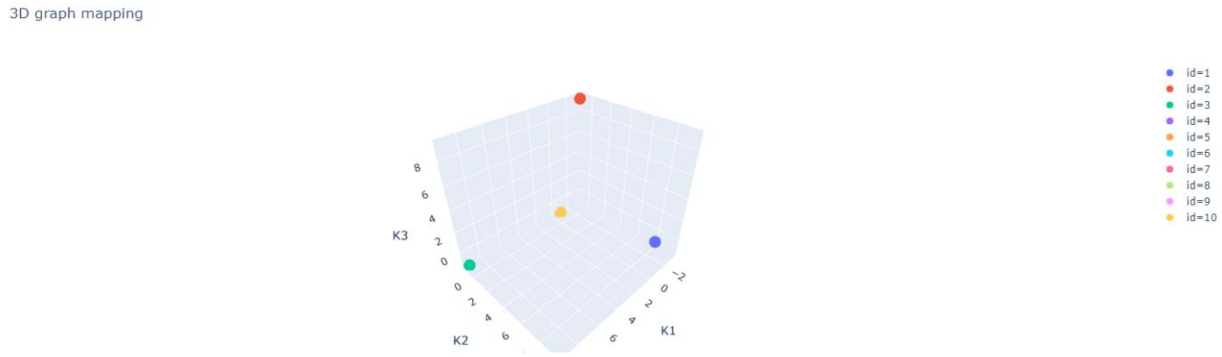

Рис. 4.4. Проекция графа K_{10} на базис, полученный из собственных векторов, соответствующих 3 минимальным ненулевым собственным числам (в данном случае, это не привнесло интерпретируемости).

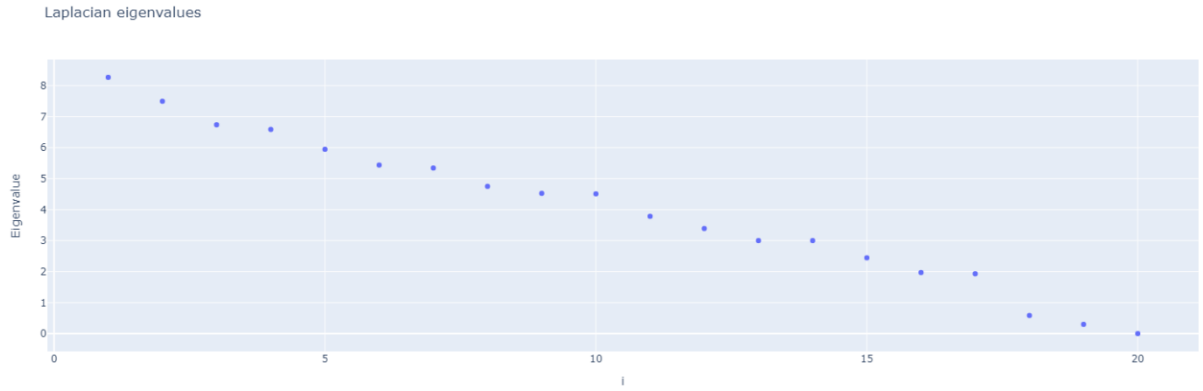

Рис. 4.5. Собственные вектора Лапласиана графа G_2. Ожидаемо, арифметическая связность ненулевая, поскольку граф является связным.

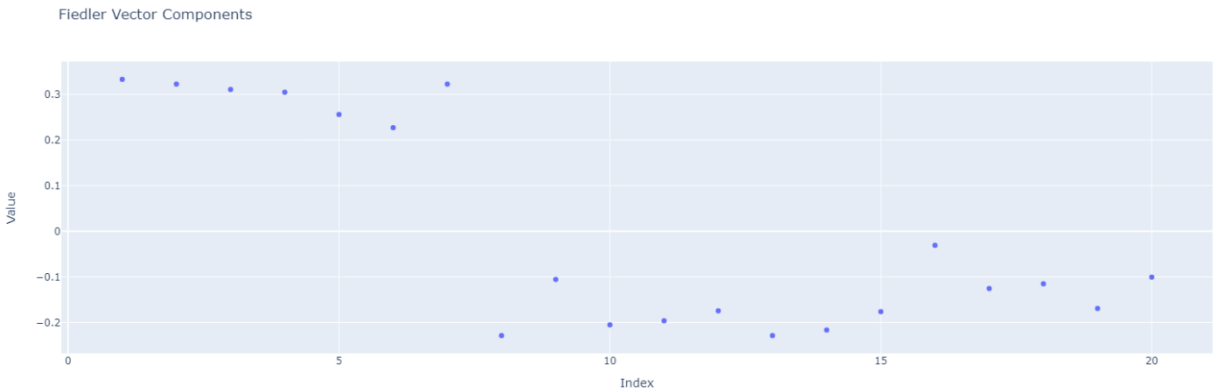

Рис. 4.6. Компоненты вектора Фидлера для графа G_2.

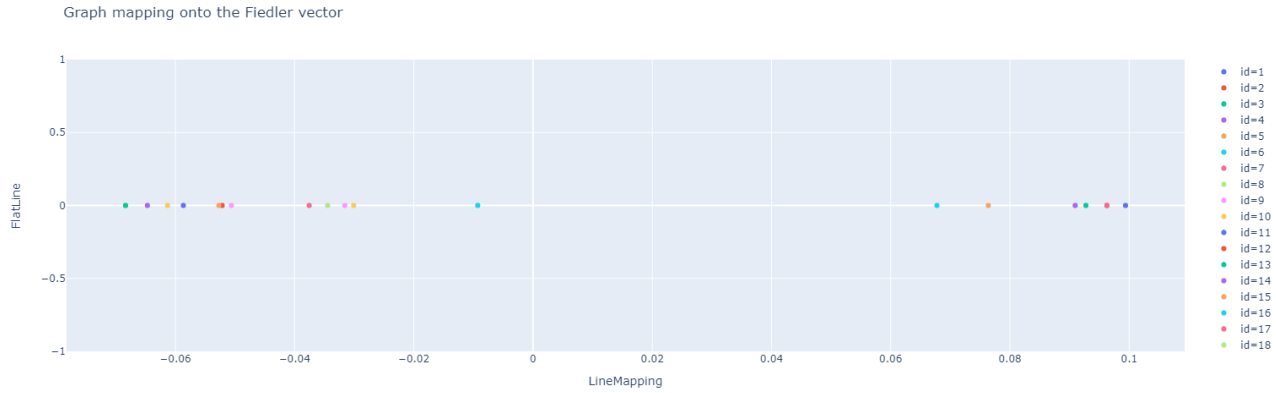

Рис. 4.7. Проекция G_2 на вектор Фидлера. Даже в одномерном случае можно наблюдать 3 «скопления» вершин, соответствующих 3 кластерам.

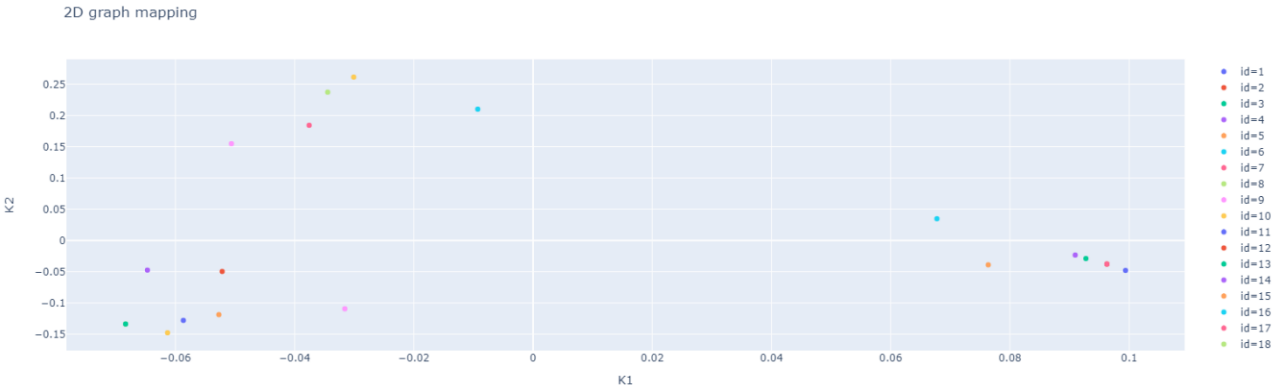

Рис. 4.8. Проекция G_2 на двумерное пространство. Кластеры становятся ещё более различимыми.

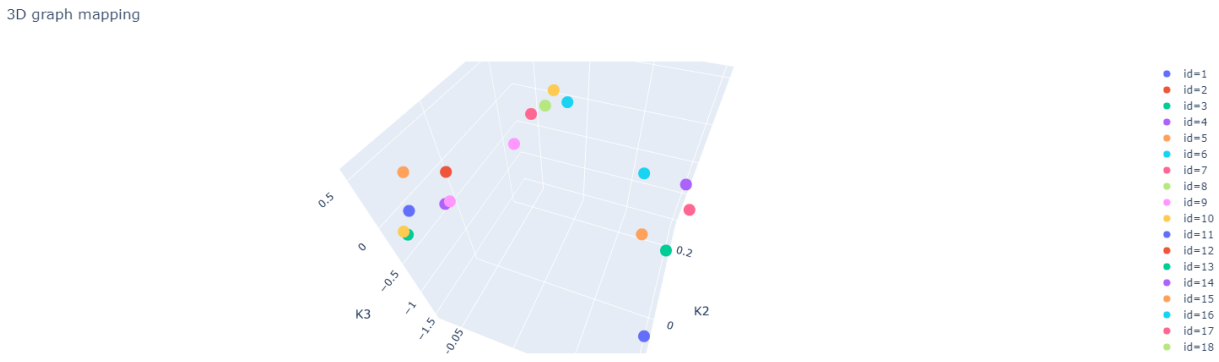

Рис. 4.9. Отображение G_2 на трехмерное пространство.

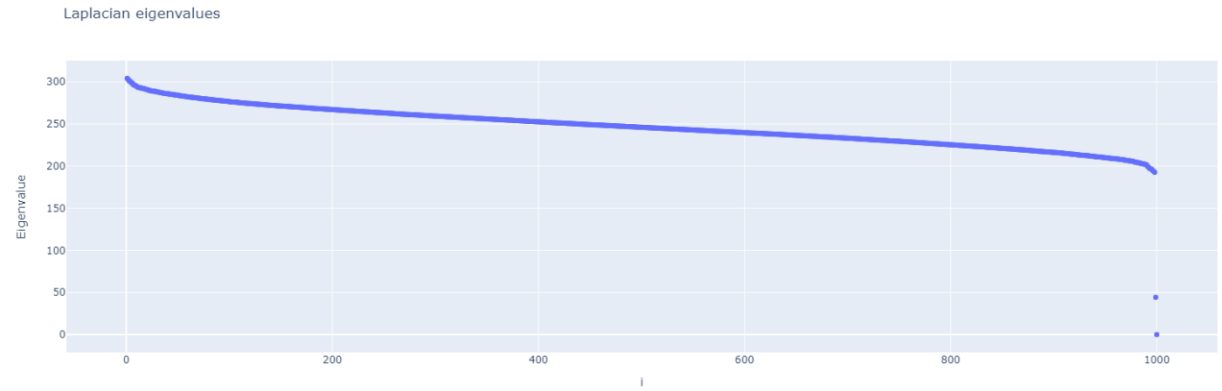

Рис. 4.10. Собственные числа Лапласиана для G_3.

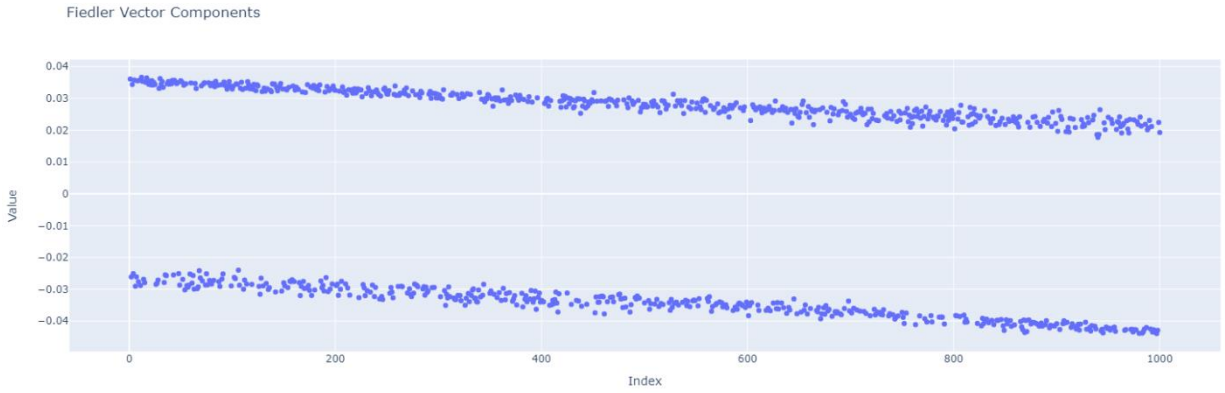

Рис. 4.11. Значения компонент вектора Фидлера для G_3. Можно наблюдать, что близких к 0 компонент не наблюдается, в графе 2 ярко выраженных кластера (причем один больше другого, это заметно по плотности точек).

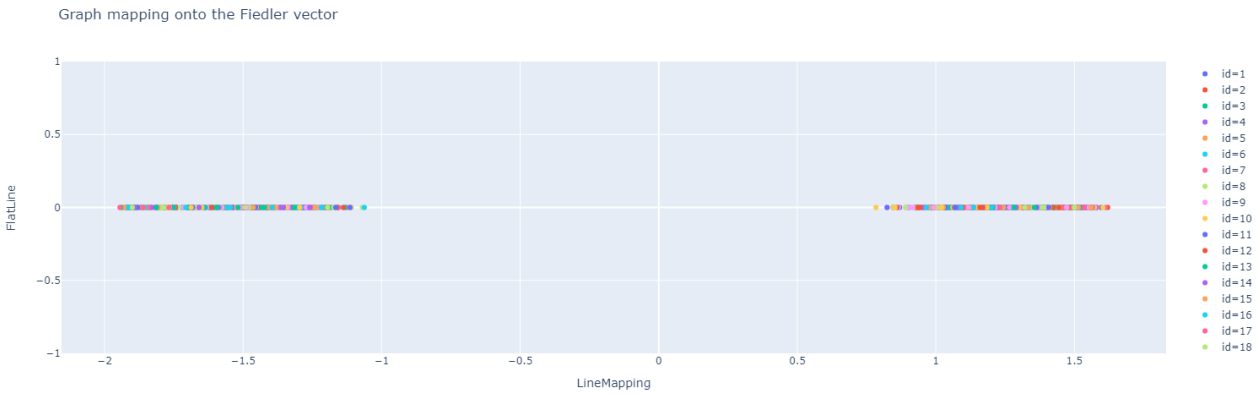

Рис. 4.12. Проекция G_3 на вектор Фидлера. 2 заметно выраженных кластера.

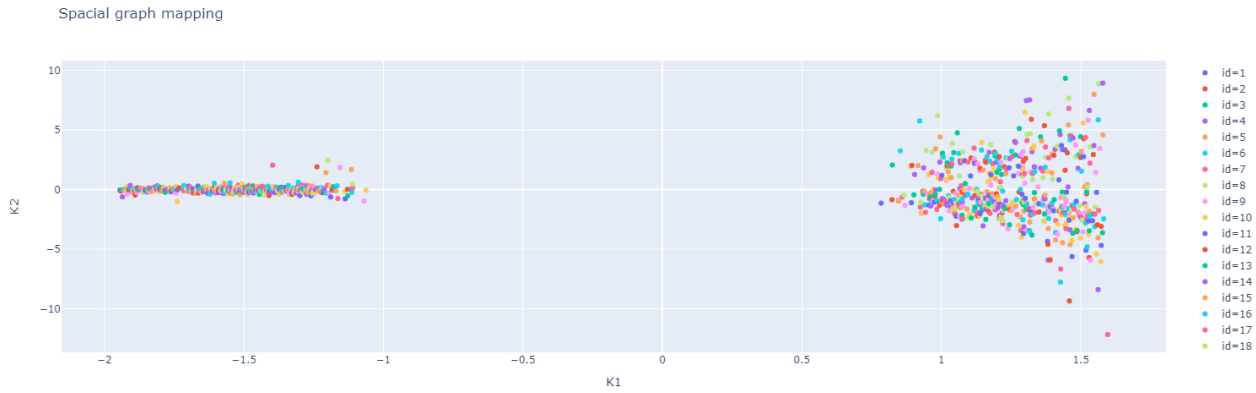

Рис. 4.13. Проекция G_3 на двухмерное пространство. Часть элементов, имеющая сильное отклонение вдоль оси ординат, была скрыта. Теперь становится очевидно, что левый кластер плотнее правого, проекции его вершин на 2 компоненту меньше по абсолютному значению.

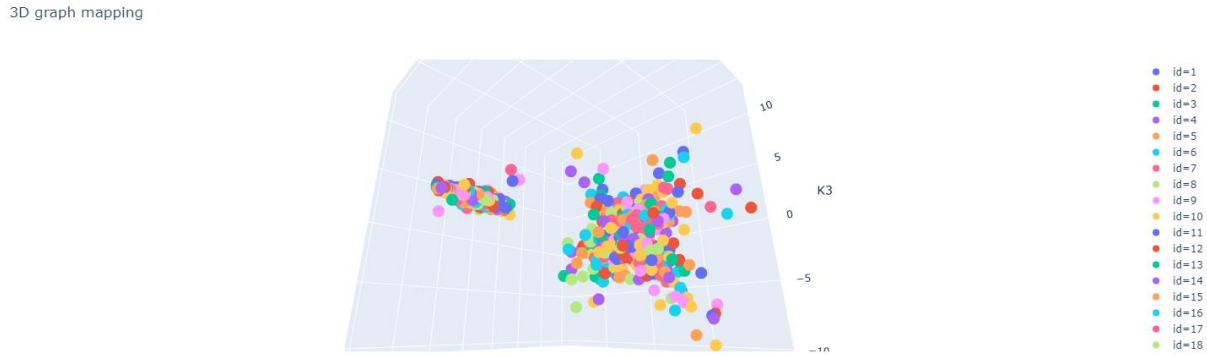

Рис. 4.14. Отображение G_3 на трехмерное пространство, полученная аналогичным образом. Как и в случае двумерного пространства, правый кластер многочисленнее левого, но при этом плотность элементов в нем ниже.

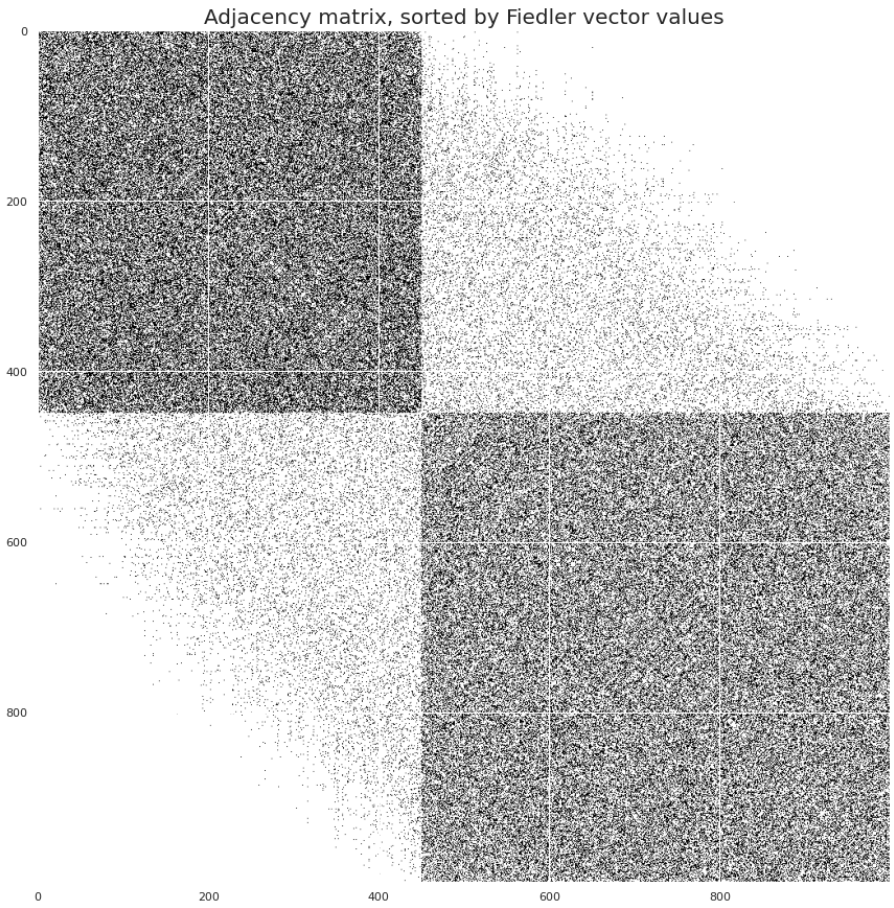

Рис. 4.15. Матрица смежности G_3, отсортированная по возрастанию значений вектора Фидлера. Можно наблюдать 2 ярко выраженных плотных кластера, при этом верхний левый визуально более темный, что связано с большей плотностью черных точек, соответствующих единицам. Количественно граница кластеров проходит примерно на уровне 450 вершины, то есть левый верхний кластер содержит меньше вершин.

В качестве опционального задания предлагалось произвести кластеризацию графа G_2 методом DBSCAN (сокр. Density Based Spatial Clustering for Applications with Noise, англ. Алгоритм плотностной пространственной кластеризации для приложений с шумом). Как и упоминается в названии, данный алгоритм определяет кластеры, основываясь на распределении плотности точек в многомерном пространстве. В качестве меры близости обычно принимается евклидово расстояние. Изначально алгоритм определяет некоторые опорные элементы на основании близости с заданным числом точек. После этого производит поиск непомеченных точек в некоторой окрестности ϵ каждой из помеченных точек. Если в определенный момент ни одной точки не было найдено в окрестности текущих помеченных точек, производится выбор нового опорного элемента и процесс повторяется. Все изолированные опорные элементы помещаются в зарезервированный кластер с номером -1, то есть помечаются как шум.

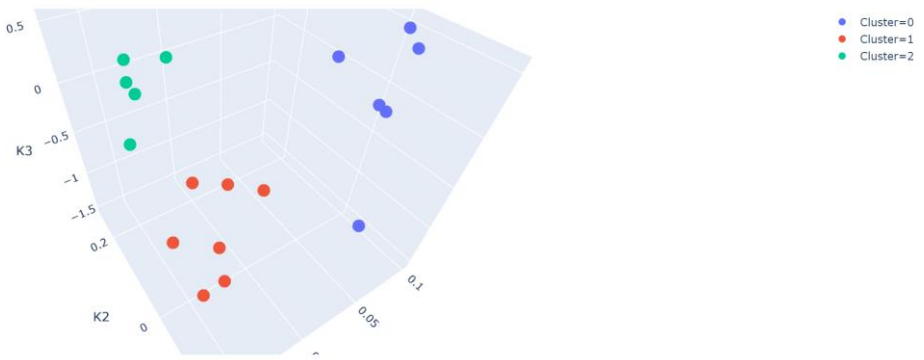

Рис. 4.16. Результат применения алгоритма DBSCAN к G_2: корректно определены 3 кластера.

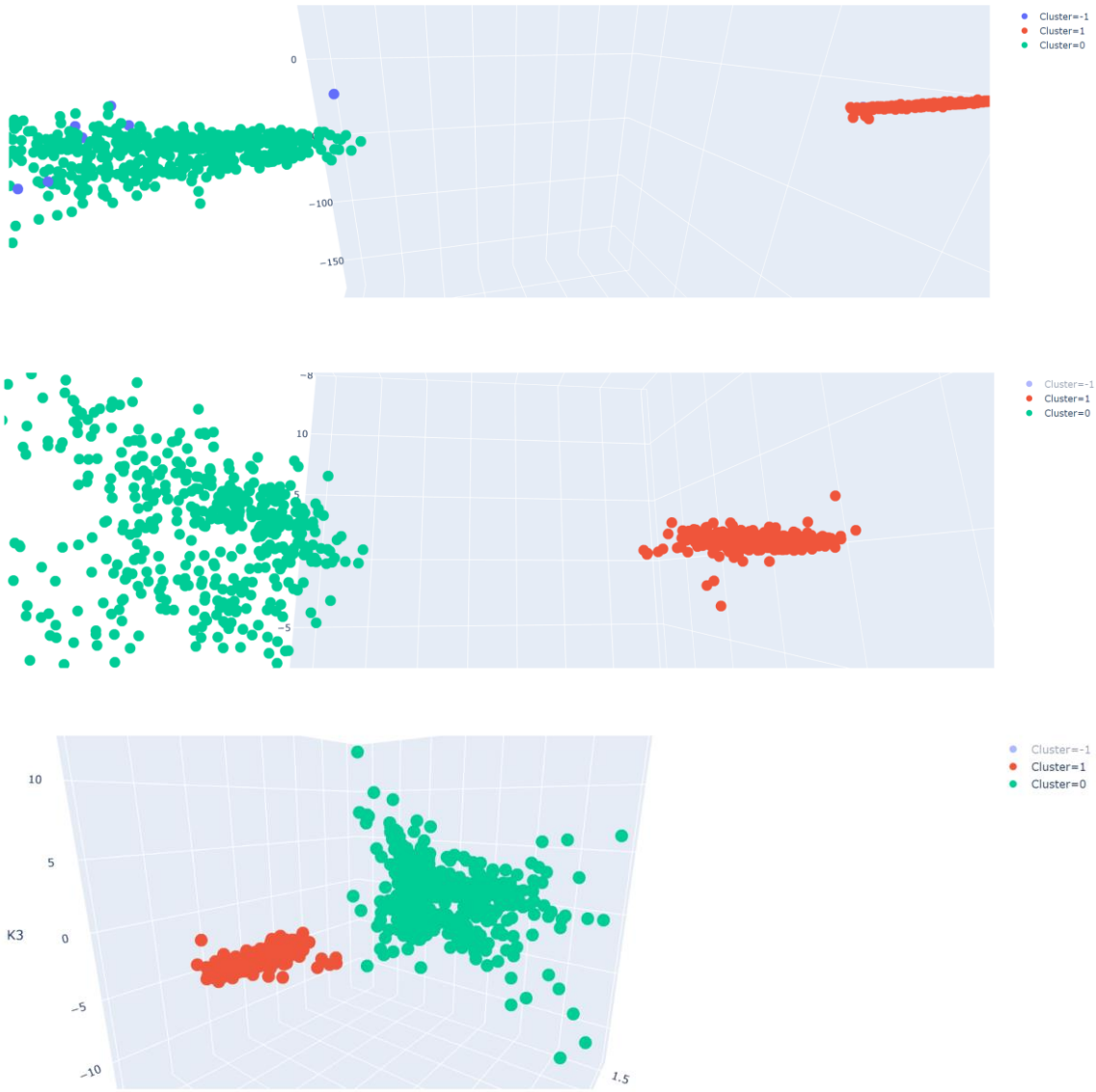

Рис. 4.17. Результат применения алгоритма DBSCAN к G_3: корректно определены 2 кластера (красная и зеленая области), некоторые вершины попали в кластер с номером -1, т. е. они являются изолированными (синяя область).

Заключение

Численные методы линейной алгебры имеют очень широкое применение, поскольку позволяют решать многие типовые задачи, возникающие при работе с многомерными данными. В рамках данной работы был рассмотрен такой важный метод понижения размерности, как метод главных компонент. Уже на примере результативности линейных моделей машинного обучения, обученных на изначальных данных, нормализованных начальных данных и данных пониженной размерности можно наблюдать преимущества, которые дает такой подход.
Куда менее очевидным и сложным способом применения является спектральное разложение графов, которое может применяться для поиска числа компонент связности, разделения графа на остовые деревья и получения отображений (кодирований, *embedding*) графа в пространствах различной размерности. В основе этих, казалось бы, совсем несвязанных алгоритмов лежит одна идея, заключающаяся в разложении матриц на собственные числа и вектора. Удивительно, как эта простая абстракция может помогать решать такие различные, порой нетривиальные, задачи.

Список использованных источников

1. **Соколов А. П., Першин А. Ю.** Инструкция по выполнению лабораторных работ (общая). // кафедра «Системы автоматизированного проектирования» МГТУ им. Н. Э. Баумана, Москва, 2021.
2. **Першин А. Ю.** Лекции по вычислительной математике. // Кафедра РК6 (Системы автоматизированного проектирования) МГТУ им. Н. Э. Баумана, 2020.
3. **Higham, Nicholas J.** Accuracy, and stability of numerical algorithms // University of Manchester, Manchester, England, 2002.
4. **Lilian Dai.** Spectral Graph Theory and its Applications. Веб-ресурс. // **URL:** http://web.mit.edu/6.454/www/www_fall_2004/lldai/slides.pdf [Дата обращения: 12.12.2021]
5. **Wayne Barrett.** The Fiedler Vector and Tree Decompositions of Graphs. // Brigham Young University. Веб-ресурс. // **URL:** https://math.byu.edu/wp-content/uploads/2020/01/WayneTalks/TreeDecompositionsSaskatoon.2015.pdf [Дата обращения: 11.12.2021]