«Созревание» собственного вектора центральности гиперграфа соавторств

Насыров Руслан Рашидович
студент 1 курса,
физтех-школа прикладной математики и информатики,
Московский физико-технический институт(ГУ)
РФ, г. Долгопрудный
E-mail: nasyrov.rr@phystech.edu
Мусатов Даниил Владимирович
Кандидат физ.-мат. наук, кафедра дискретной математики МФТИ
РФ, Долгопрудный

14 мая 2021 г.

1 Аннотация

1. Объект исследования: вектор центральности гиперграфа.
2. Цель: анализ «созревания» вектора центральности гиперграфа при различных проекциях.
3. Задачи:
   (a) дать определение ключевым понятиям
   (b) определить центральность вершин гиперграфа
   (c) найти собственные векторы центральности гиперграфа и его проекций
   (d) проанализировать созревание векторов центральности
   (e) сделать вывод

2 Ключевые понятия

Определение 1 *Гиперграф - обобщенный вид графа, в котором каждым ребром могут соединятся не только две вершины, но и любые подмножества вершин. С математической точки зрения, гиперграф H - это пара H=(V, I) , где V - множество вершин, а I семейство подмножеств V, называемых гиперребрами.* [3]

Определение 2 *Размерность гиперграфа H = (V, I) - число* $dim(H) = max|h|, h \in I$. [3]

Определение 3 *k-проекция гиперграфа H = (V, I) - это гиперграф*
$\pi_k(H) = (V, J_k)$ *где* $J_k \subset I$ *и* $\forall h \in J_k : |h| \leq k$, $2 \leq k \leq dim(H)$.
В этом случае будем называть k номером проекции. [3]

Определение 4 *Матрица смежности гиперграфа H(V, I) - это такая матрица* $A \in M_{|V|\times|V|}$, *что* $\forall i, j : 1 \leq i, j \leq |V|$ *если* $A[i][j] = 1$ *то* $\exists h \in I : i, j \in h$. *Иначе A[i][j] = 0.* [3]

Определение 5 *Собственный вектор гиперграфа - такой вектор v, что* $\exists \lambda \in R : Av = \lambda v$, *где A - матрица смежности гиперграфа. В этом случае* $\lambda-$ *собственное значение гиперграфа.* [3]

Определение 6 *Собственный вектор центральности гиперграфа - собственный вектор, отвечающий максимальному собственному значению. Его существование и неотрицательность(все элементы* ≥ 0*) следует из теоремы Перрона-Фробениуса.* [2].

Определение 7 *Предельный вектор центральности гиперграфа - собственный вектор центральности всего графа.*

Определение 8 *Созревание вектора центральности гиперграфа - процесс стабилизации вектора центральности проекции гиперграфа с увеличением номера проекции.*

3 Введение

3.1 Актуальность

Гиперграфы, широко применяются для представления взаимодействия между элементами какой-либо сложной системы:

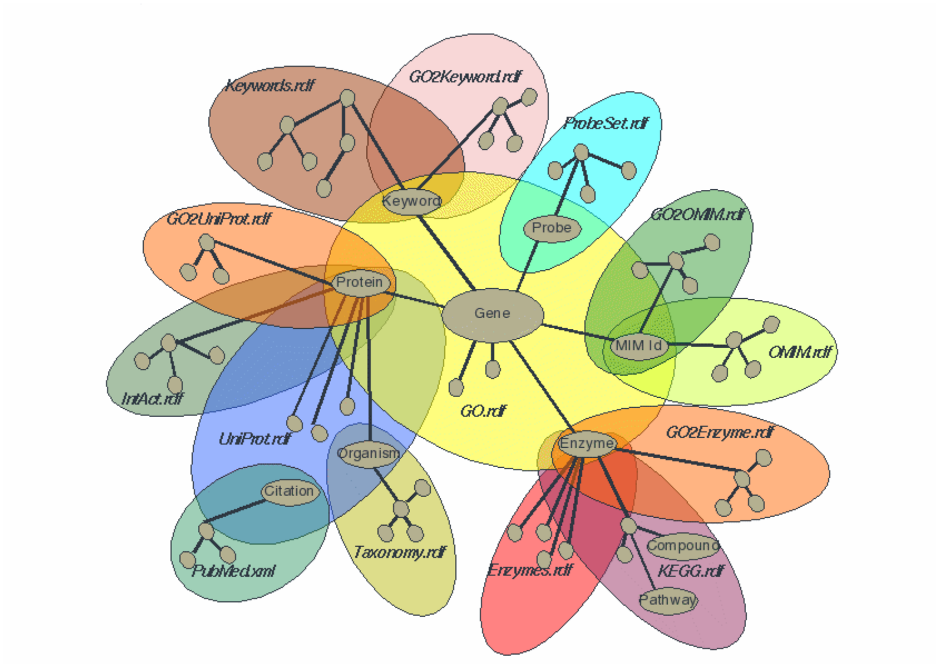

Рис. 1: Биохимическая сеть[10]

будь то трофическая сеть, социальная или биохимическая. Эти сети возникают с учетом взаимодействия между более чем двумя узлами одновременно, поэтому для их представления классические сетевые модели недостаточны.

Анализ созревания вектора центральности гиперграфа соавторств имеет значение в приложениях, поскольку раннее созревание вектора позволяет не рассматривать ребра высокой мощности, что ускоряет и упрощает дальнейший анализ гиперграфов.

3.2 Гипотеза

Предположим, что наблюдается созревание вектора центральности гиперграфа с увеличением номера проекции и происходит корреляция между предельным вектором и вектором различных проекций, и проверим это.

3.3 Научная новизна

Впервые исследуется созревание вектора центральности гиперграфа с увеличением номера проекции исходя из корреляции между ним предельным вектором. Предлагается сформировать вектор центральности новым, экспериментальным способом.

4 Центральность вершин гиперграфа

Показатель центральности определяет наиболее важные вершины графа. Он применяется для выявления наиболее влиятельного лица в социальной сети, ключевых узлов инфраструктуры в интернете или городских сетей и разносчиков болезни.

Один из известных подходов к определению центральности вершины состоит в том, что центральность вершины должна быть прямо пропорциональна центральности ее соседей, причем можно использовать различные функции пропорцинальности, не обязательно линейные.

Пусть $i \in V$, центральность этой вершины c(i) определим так:

$$c(i) = \frac{1}{\lambda} \sum_{k=2}^{n} \sum_{i, j_1, ..., j_{k-1} \in I} F_k(c(j_1), ..., c(j_{k-1})$$

Функция F может быть разной, соответственно возникает интерес исследовать созревание при различных функциях F и сравнить результаты.

1. $F(x_1, ..., x_{k-1}) = x_1 + ... + x_{k-1}$
   Если использовать такую линейную функцию, то получится, что вектор центральности - собственный вектор. Именно она и будет применяться для дальнейшего исследования.

2. $F(x_1, ..., x_{k-1}) = x_1 ... x_{k-1}$

   Если рассматривать такую нелинейную функцию, то получится, что вектор центральности - Z-собственный вектор, который рассматривается в статье A.R.Benson[1].

3. $F(x_1, ..., x_{k-1}) = ||(x_1, ..., x_{k-1})||_p, p \in R$

Для всех этих функций существование и единственность вектора центральности задается нелинейной теорией Перрона-Фробениуса [2].

Теперь, когда центральность вершин гиперграфа определена, существенный интерес представляет исследование зависимости между вектором центральности всего гиперграфа и его проекций.

5 Нахождение собственного вектора центральности гиперграфа и его проекций

Будем рассматривать гиперграфы, ребра которых - это научные статьи, вершины - это авторы, которые принимали участие в написании статей по направлению Computer Science.

План исследования:

1. Выгрузка данных из источника.

2. Создание гиперграфа.

3. Вычисление проекций гиперграфа.

4. Вычисление векторов центральности этих проекций.

5.1 Выгрузка данных

Выгрузка данных из источника datasets[7] была произведена с использованием библиотеки json[8].

5.2 Создание гиперграфа

Гиперграф создан из списка ребер с использованием библиотеки networkx[6]. Его характеристики:

1. Вершин: 363043

2. Гиперребер: 435135

3. Максимлальное гиперребро: 427 вершин

4. Средний размер ребра: 4
5. Компонент связности: 22912
6. Максимальный размер компоненты: 298870
7. Средний размер компоненты: 16. В основном, размеры компонент - из отрезка [1, 61]

   На основании этих данных можно заключить, что граф - довольно разреженный, около 80% вершин содержатся в одной большой компоненте. Так что имеет смысл отбросить мелкие компоненты и анализировать только большую

5.3 Вычисление проекций гиперграфа

```
def get_projections_to_graph(path_to_json, path_to_save_proj):
    file = open(path_to_json, 'r', encoding='utf-8')
    data = json.load(file)
    file.close()
    edges = data.values()
    names = list(set([name for i in edges for name in i]))
    n = len(names)
    n_nodes = n
    name_to_int = dict({names[i]: i for i in range(len(names))})

    numerical_edges = []
    for edge in edges:
        lst = []
        for name in edge:
            lst.append(name_to_int[name])
        numerical_edges.append(np.array(lst))
    numerical_edges = sorted(numerical_edges, key=lambda x: len(x))

    def save_gr_proj(G, num):
        full_p = path_to_save_proj + "\\proj # " + str(num) + ".txt"
        nx.write_edgelist(G, full_p)


    G = nx.Graph()
    G.add_nodes_from(np.arange(n))

    prs = 0
    for he in tqdm(numerical_edges):
        sz = len(he)
        if sz != prs:
            if prs >= 2:
                save_gr_proj(G, prs)
            print(prs, ' done')
            prs = sz
        for id1 in range(len(he)):
            for id2 in range(id1 + 1, len(he)):
                G.add_edge(he[id1], he[id2])

    save_gr_proj(G, prs)
```

Код 1: вычисление проекций гиперграфа.

Проекции вычислены, количество проекций гиперграфа - 78 штук.

5.4 Вычисление векторов центральности проекций гиперграфа

```
def get_centrality_to_graph(path_to_proj, path_to_save_centr):
    for numb in tqdm(numbers_projections):
        full_p = path_to_proj + "\\proj # " + str(numb) + ".txt"
        G = nx.read_edgelist(full_p)
        print('edg: ', G.number_of_edges())
        e = G.number_of_nodes()
        print(e, n_nodes)
        G.add_nodes_from(str(x) for x in np.arange(n_nodes))
        vector = np.array(nx.eigenvector_centrality_numpy(G, 100).values())
        file = open(path_to_save_centr + "\\centr_for_proj # " + str(numb) + ".txt", 'w')
        file.write(str(vector)[13:-2])
        file.close()
```

Код 2: вычисление векторов центральности проекций гиперграфа.

Векторы центральности вычислены с использованием библиотеки networkx[6] по методу Перрона–Фробениуса. [2]

6 Анализ созревания вектора центральности

План анализа:

1. Расчёт корреляции между предельным вектором центральности и векторами центральности проекций графа.
2. Построение графика корреляции.
3. Вывод.

Рассмотрим корреляцию топ-n вершин предельного вектора и вектора проекций при

1. n = 100
2. n = 1000
3. n = 363043 (все вершины)

Для этого:

1. Занумеруем в предельном векторе вершины в порядке убывания их центральности.
2. Возьмем первые n вершин, так получим предельный вектор.
3. Для каждой проекции попробуем сформировать вектор следующим экспериментальным способом: если вершина v в k-ой проекции встречается среди топ-n вершин и предельного вектора, и вектора своей проекции, то в результирующем векторе ей назначается ранее вычисленная центральность, иначе - 0.

6.1 Вычисление корреляции

```python
def get_corr_vector(n, path_to_perm, path_to_centrality):

    def get_numbers_of_top_n_in_decreased_order_of_centrality(n, fileperm):
        f = open(fileperm, "r")
        line = f.readline()
        f.close()
        lst = np.array(list(map(int, line.split(', '))))[:n]
        return lst

    def get_sorted_numbers_of_top_n(n, fileperm):
        return np.array(sorted(get_numbers_of_top_n_in_decreased_order_of_centrality(n, fileperm)))

    def get_centrality_vector(filecentr):
        file = open(filecentr, 'r')
        line = file.readline()
        file.close()
        vector = np.array(list(map(float, line.split(', '))))
        return vector


    def get_centr_of_ids(n, ids): # centrality for all projections
        ans = []
        for numb in numbers_projections:
            file_centr = path_to_centrality + "\\centr_for_proj # " + str(numb) + ".txt"
            file_perm = path_to_perm + "\\" + "\\num_center_for_proj # " + str(numb) + ".txt"

            vector = get_centrality_vector(file_centr)
            top_n = get_sorted_numbers_of_top_n(n, file_perm)

            peres = np.intersect1d(top_n, ids)

            temp = []
            for i in ids:
                if i in peres:
                    temp.append(vector[i])
                else:
                    temp.append(0)

            ans.append(np.array(temp))
        #by rows: number of projection
        #by columns: number of vertex
        return np.array(ans)

    def get_centr_of_top_n(n): #n from back(full graph)
        idx = get_sorted_numbers_of_top_n(n, path_to_perm + "\\" + \
                                          "num_center_for_proj # " + str(numbers_projections[-1]) + ".txt")
        return get_centr_of_ids(n, idx)


    centr_n = get_centr_of_top_n(n)

    table_n = pd.DataFrame(centr_n)
    table_n.index = numbers_projections
    table_n.columns = ["ver# " + str(x) for x in np.arange(n)]
    table_n = table_n.T
    limit_vector = table_n[table_n.columns[-1]]
    corr_n_centr_between_last = []
    for i in range(table_n.shape[1]):
        tau, _ = stats.kendalltau(table_n[table_n.columns[i]], limit_vector)
        corr_n_centr_between_last.append(tau)
    return corr_n_centr_between_last
```

Код 3: вычисление корреляции между векторами гиперграфа.

Корреляция между предельным вектором и вектором проекции была вычислена с помощью функции scipy.stats.kendalltau библиотеки scipy.[9] Более подробно данная функция корреляции рассмотрена в статье Kendall. [4]

6.2 Построение графиков

```python
def draw_graphics(correl, n):
    fig, ax = plt.subplots(figsize=(15, 15))
    ax.set_title("Корреляция между вектором центральности и предельным\
                                                    вектором(топ-{})".format(n), fontsize=20)
    ax.set_xlabel("номер проекции", fontsize=20)
    ax.set_ylabel("корреляция", fontsize=20)


    ax.plot(numbers_projections[1:-1], correl[1:-1], c='r')
    ax.scatter(x=numbers_projections[1:-1], y=correl[1:-1], marker='X', s=100, label="проекция")
    ax.legend()
```

Код 4: построение графиков корреляции между векторами гиперграфа.

Графики были построены с использованием библиотеки matplotlib[5].

6.3 n=100

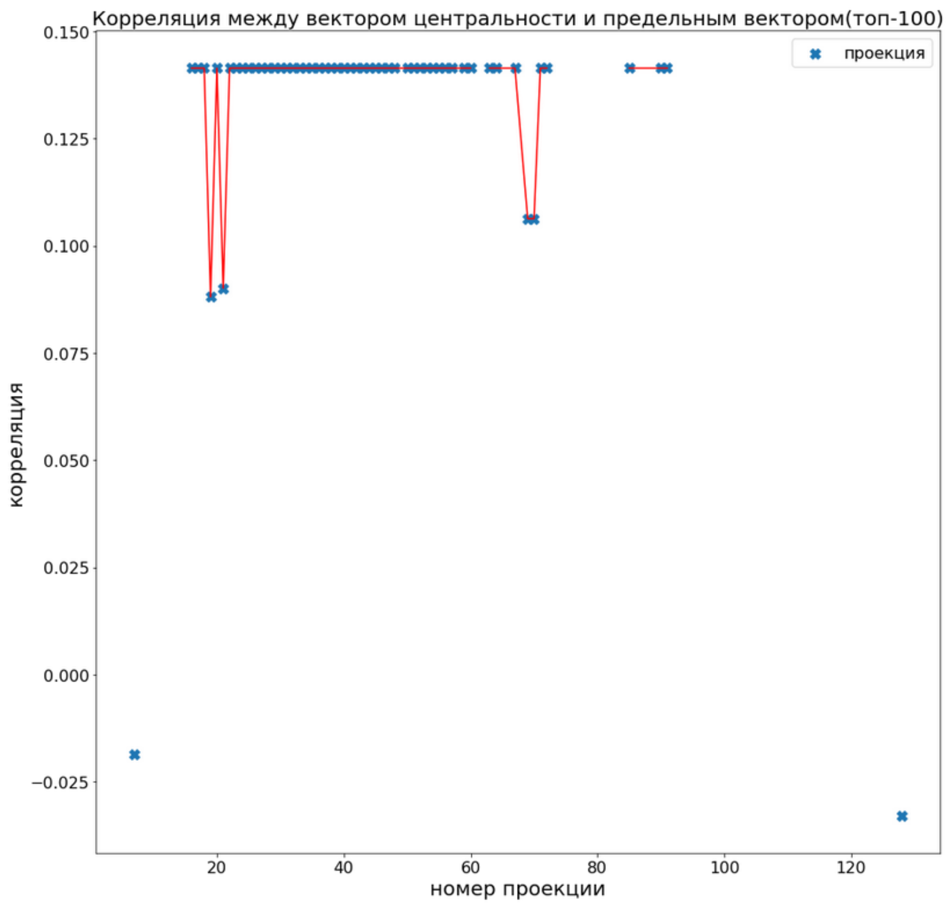

Рис. 2: График корреляции между векторами центральности проекций топ-100 вершин

Видим: векторы проекций слабо скоррелированы с предельным вектором. Кроме того, на ранних проекциях (под омерами 20 - 90) корреляция с предельным вектором почти одинаковая(около 0.140). Отсутствие точек для какой-то проекции означает, что топ-100 вершин этой проекции не пересекается с топ-100 предельного вектора. Соответственно, в этом случае нет смысла говорить о корреляции.

6.4 n=1000

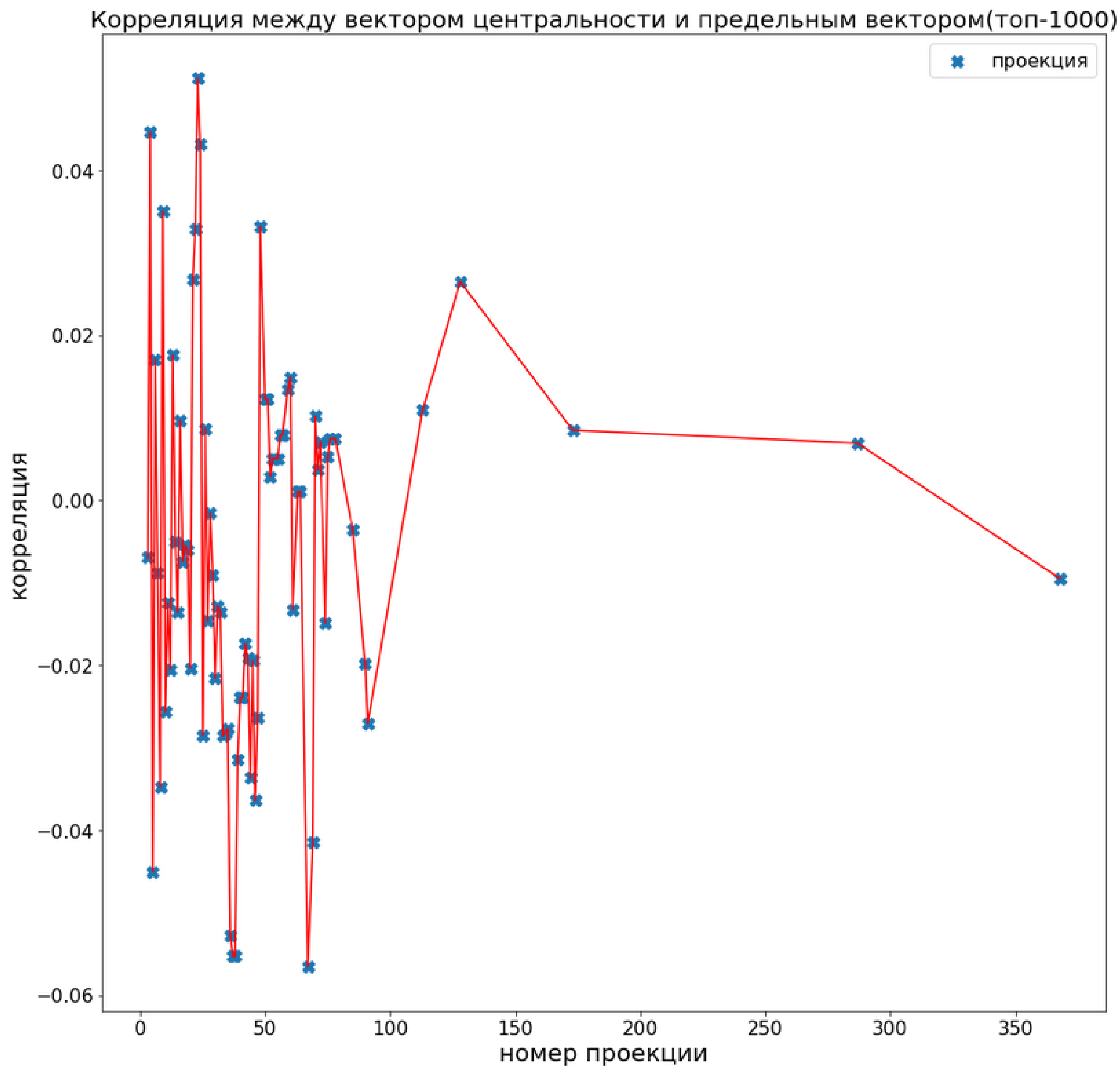

Рис. 3: График корреляции между векторами центральности проекций топ-1000 вершин

Видим: корреляции нет.

6.5 Для всех вершин

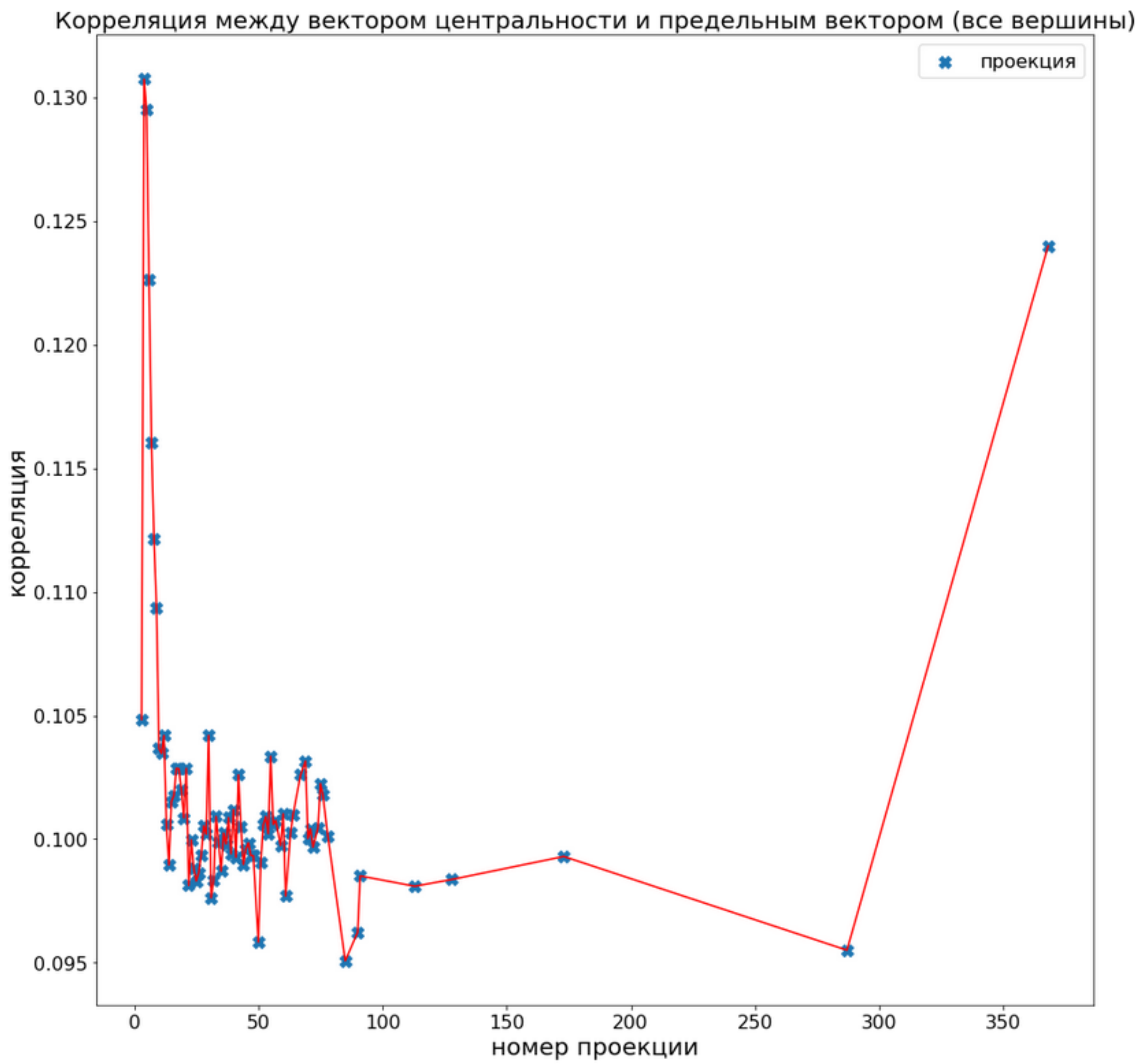

Рис. 4: График корреляции между векторами центральности проекций всех вершин

Видим: корреляции почти нет.

6.6 Вывод

Вектор центральности гиперграфа сильно изменяется при добавлении больших по мощности гиперребер и слабо при добавлении малых(на ранних проекциях). При топ-1000 и для всех вершин корреляция сильно скачет на начальных проекциях. Созревания вектора центральности гиперграфа при различных проекциях не наблюдается.

7 Итог

В рамках исследования была написана программа позволяющая анализировать созревание вектора центральности для произвольного гиперграфа. С её помощью проанализировано созревание вектора центральности для гиперграфа статей по направлению Computer Science. На основе полученных результатов можно сделать вывод, что созревания вектора центральности гиперграфа не наблюдается и наша гипотеза не подтвердилась.

Впоследствии в рамках исследования будет изучено созревание вектора при нелинейных функциях F из части 4 и будет изучено созревание относительно стабилизации топ-n вершин (то есть начиная с какого номера проекции топ-n самых зачимых по центральности вершин перестанет изменяться).

Список литературы

[1] A.R. Benson, "Three hypergraph eigenvector centralities SIAM Journal on Mathematics of Data Science 1(2) (2019), 293312.

[2] B. Lemmens and R. Nussbaum, "Nonlinear Perron-Frobenius Theory CambridgeTracts in Mathemtics 189 (2012). Cambridge University Press.

[3] В. А. Емеличев, О. И. Мельников, В. И. Сарванов, Р. И. Тышкевич. Глава XI: Гиперграфы // Лекции по теории графов. — М.: Наука, 1990. — С. 298—315. — 384 с. — ISBN 5-02-013992-0.

[4] Maurice G. Kendall, "The treatment of ties in ranking problems", Biometrika Vol. 33, No. 3, pp. 239-251. 1945.

[5] matplotlib.org

[6] networkx.org

[7] github.com/mattbierbaum/arxiv-public-datasets

[8] json.org

[9] scipy.org

[10] medium.com/@jeffreystewart/semantic-data-master