Федеральное государственное автономное
образовательное учреждение высшего образования
Национальный исследовательский университет
«Высшая школа экономики»

Факультет математики

Беребердина Наталья Александровна

Геометрическое машинное обучение с помощью потока Риччи: выявление скрытых структур в сложных наборах данных

Дипломная работа студентки 4-го курса
образовательной программы бакалавриата «Математика»

Научный руководитель:
Тупикина Любовь Владимировна
ИТМО, МФТИ
Кандидат физико-математических наук

Москва 2025

Содержание

1 Введение

В этой работе мы строим метрические графы из наборов данных MNIST и ArXiv, вычисляем кривизну Оливье-Риччи на ребрах, применяем алгоритм потока Риччи и выполняем удаление ребер. С помощью таких манипуляций мы получаем новые графы, анализируя которые можно получить новую информацию об исходных данных. Исследование мотивировано взаимосвязью между законами масштабирования нейросетей и фундаментальными геометрическими свойствами данных. Законы масштабирования нейросетей описывают, как точность нейронных сетей предсказуемо улучшается с увеличением размера модели, объема данных или вычислительных ресурсов. Одно из важных предположений в этой области заключается в том, что эти законы масштабирования тесно связаны с внутренней структурой данных.

Выявляя эти структуры, мы потенциально можем оптимизировать процесс обучения — например, сосредоточившись на репрезентативных подмножествах данных или динамически адаптируя архитектуру сети для лучшего соответствия топологии данных. Поток Риччи, который делает кривизну графа постоянной, является, по нашему предположению, одним из способ выявления таких структур. В частности, ребра с высокой кривизной после потока Риччи могут соответствовать границам между различными кластерами, а ребра с низкой кривизной могут указывать на плотные, однородные области.

Наша основная гипотеза, которую мы стремились проверить в этой работе, связана с кластеризацией. Мы предполагаем, что срезка ребер с большим весом после потока Риччи помогает найти сообщества в графе, которые в свою очередь соответствуют определенным кластерам в реальных данных.

В конечном итоге это исследование пытается ответить на следующий вопрос: Что мы можем узнать о структуре данных, анализируя кривизну на разных этапах потока Риччи?

2 Вычисление потока Риччи

2.1 Кривизна Оливье-Риччи

Определение 1 (Вероятностная мера вершины для взвешенного графа)**.** *Для вершины $v \in V$ во взвешенном графе $G = (V, E, w)$, где $w : E \to \mathbb{R}^+$ задаёт положительные веса рёбер:*

$$m_v(x) := \begin{cases} \dfrac{w(v,x)}{\sum\limits_{u \in \mathcal{N}(v)} w(v,u)} & \text{если } x \in \mathcal{N}(v), \\ 0 & \text{иначе,} \end{cases}$$

где $\mathcal{N}(v)$ — окрестность вершины v, а $w(v, x)$ обозначает вес ребра.

Определение 2 (Расстояние Вассерштейна-1)**.** *Пусть μ, ν — вероятностные меры на метрическом пространстве (V, d).* Расстояние Вассерштейна-1 *(или расстояние землекопа) между ними:*

$$W_1(\mu, \nu) := \infty_{\gamma \in \Pi(\mu,\nu)} \sum_{(x,y) \in V \times V} d(x, y)\gamma(x, y),$$

где $\Pi(\mu, \nu)$ обозначает множество всех сопряжений μ и ν.

Определение 3 (Кривизна Оливье-Риччи)**.** Кривизна Оливье-Риччи *ребра $(x, y) \in E$:*

$$\kappa(x, y) := 1 - \frac{W_1(m_x, m_y)}{d(x, y)},$$

где $d(x, y)$ — расстояние в графе (обычно $d(x, y) = 1$ для смежных вершин).

2.2 Поток Риччи на графе

Поток Риччи был впервые введён Ричардом Гамильтоном в 1982 году [5] как инструмент исследования римановых метрик на многообразиях. Уравнение потока имеет вид:

$$\frac{\partial g_{ij}}{\partial t} = -2\mathrm{Ric}_{ij},$$

где g_{ij} - метрический тензор, а Ric_{ij} - тензор Риччи. В той же работе Гамильтон доказал, что на трёхмерных многообразиях с положительной кривизной Риччи поток приводит к метрике постоянной положительной кривизны. В 2000-е годы началась адаптация потока Риччи для графов.

Определение 4 (Поток Риччи для взвешенного графа)**.** *Для взвешенного графа $G = (V, E, w)$ с весами рёбер $w_t : E \to \mathbb{R}^+$, зависящими от параметра t, поток Риччи задаётся дифференциальным уравнением:*

$$\frac{d}{dt} w_t(v_i, v_j) = -\kappa(v_i, v_j) \cdot w_t(v_i, v_j)$$

где:

- $\kappa(v_i, v_j)$ — *кривизна Риччи ребра* (v_i, v_j) *в момент* t
- $w_t(v_i, v_j)$ — *вес ребра* (v_i, v_j) *в момент времени* t

Дискретный алгоритм потока Риччи для взвешенных графов является итеративным алгоритмом, зависящим от времени. На каждой итерации поток Риччи генерирует взвешенный граф $(V, E, w(t))$. Изменения веса каждого ребра зависят от кривизны Оливье-Риччи. Дискретная реализация [3] использует правило обновления:

$$w_{ij}(t+1) = (1 - \eta\kappa_{ij}(t))d^t(i, j), \tag{1}$$

где $d^t(i, j)$ — кратчайшее расстояние между вершинами i и j с учетом весов в момент времени t, $\kappa_{ij}(t)$ — кривизна ребра между вершинами i и j Оливье-Риччи, а η - параметр регулирующий скорость изменений. Здесь и далее, говоря о кривизне, мы будем иметь в виду именно определение Оливье-Риччи, хотя это только один из многих способов определить кривизну на графе. Поток Риччи можно считать для любой $f : E \to \mathbb{R}$, однако именно кривизна Оливье-Риччи дает свойства, важные для связи дискретного потока Риччи на графе и потока Риччи на многообразии.

2.3 Свойства потока Риччи

Исследуя поведение потока Риччи на графах, можно выделить несколько важных свойств, демонстрирующих его сходство и различия с потоком на многообразиях.

В отличие от непрерывного случая, где поток Риччи может вызывать изменения топологии, дискретная версия обладает свойством сохранения:

- **Связности графа**: При условии $w_{ij}(t) > 0$ для всех $t \geq 0$ и $(i, j) \in E$, граф сохраняет свою связность в процессе эволюции. Это следует из того, что поток никогда не обнуляет вес существующего ребра за конечное время.
- **Симметрии весов**: Соотношение $w_{ij}(t) = w_{ji}(t)$ сохраняется при эволюции, так как кривизна Риччи $\kappa(i, j)$ симметрична по определению.

Аналогично непрерывному случаю, существуют функционалы, монотонно убывающие вдоль потока. Для кривизны Оливье-Риччи функционал энергии $\mathcal{E}(G) = \sum_{(i,j)\in E} w_{ij}\mathrm{Ric}(i,j)$ удовлетворяет неравенству $\frac{d\mathcal{E}}{dt} \leq 0$. Эта монотонность отражает тенденцию системы к упрощению геометрической структуры.

Известно, что для некоторых классов графов поток сходится к стационарным решениям за конечное время. Например [7], для звездчатого графа с более чем двумя вершинами доказано, что поток Риччи равномерно распределяет веса рёбер, уменьшая разницу между максимальными и минимальными значениями и в пределе все рёбра приобретают одинаковый вес, что соответствует метрике с постоянной кривизной. Также доказано существование и уникальность решений для начальной задачи на произвольных временных интервалах при условии, что каждое ребро остаётся кратчайшим путём между своими вершинами [2]. В более общем дискретном случае, без ограничений на структуру графа, сходимость потока Риччи к метрике постоянной кривизны остаётся предметом активных исследований и не имеет полного решения.

Теперь перечислим некоторые экспериментальные наблюдения о потоке Риччи. Так, при достаточно большом числе итераций потока наблюдаются следующие изменения веса:

1. если $\kappa_e > 0$: ребро сжимается ($w_e \downarrow$)
2. иначе если $\kappa_e < 0$: ребро расширяется ($w_e \uparrow$)

Другие важные свойства:

1. При большом числе итераций обычно достигается состояние постоянной кривизны.
2. Ребра с большим весом обычно разделяют различные сообщества в графе.
3. Ребра с малой положительной кривизной и ненулевым весом обычно являются ребрами в плотном подграфе (сообществе).

3 Постановка задачи

3.1 Кластеризация и классификация

Кластеризация (англ. *clustering*) — задача разбиения множества объектов $\mathcal{X} = \{x_1, \ldots, x_n\}$ на k непересекающихся подмножеств (кластеров) $\mathcal{C} = \{C_1, \ldots, C_k\}$ так, чтобы объекты внутри одного кластера были «ближе» друг к другу, чем к объектам из других кластеров. Формально, минимизируется функционал:

$$\sum_{i=1}^{k} \sum_{x\in C_i} d(x, \mu_i),$$

где d — метрика расстояния, а μ_i — центр кластера C_i.

Классификация (англ. *classification*) — задача отнесения объекта $x \in \mathcal{X}$ к одному из заранее определённых классов $\mathcal{Y} = \{y_1, \ldots, y_m\}$ на основе признакового описания. Математически, ищется функция $f : \mathcal{X} \to \mathcal{Y}$, минимизирующая ошибку:

$$\mathcal{L}(f) = \sum_{(x,y)\in\mathcal{D}} \mathbb{I}(f(x) \neq y),$$

где $\mathcal{D}$ — размеченная обучающая выборка, а $\mathbb{I}$ — индикаторная функция.

3.2 Поиск сообществ

Поиск сообществ на графе (англ. *community detection*) — задача выделения подграфов $\mathcal{G}_1, \ldots, \mathcal{G}_k$ в графе $G = (V, E)$, где вершины внутри одного сообщества связаны плотнее, чем с вершинами других сообществ. Нет строгого определения лучшего разбиения на сообщества, поэтому для оценки качества разбиений вводят разные метрики, такие как коэффициент покрытия, Adjusted Rand Index, модулярность и другие. Мы будем пользоваться последним, показывающем насколько лучше наше разбиение по сравнению со случайным распределением рёбер. Это один за наиболее частых методов оценивания.

Определение 5 (Модулярность графа)**.** *Для графа $G = (V, E)$ с матрицей смежности A и заданным разбиением вершин на сообщества модулярность Q вычисляется по формуле:*

$$Q = \frac{1}{2|E|} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2|E|} \right) \delta(C_i, C_j), \tag{2}$$

- $|E|$ *— общее количество рёбер в графе,*
- d_i *— степень вершины i,*
- $\delta(C_i, C_j)$ *— индикаторная функция, равная 1, если вершины i и j принадлежат одному сообществу, и 0 в противном случае*

Кластеризация — это задача разделения набора данных на группы (кластеры) таким образом, что объекты в одном кластере похожи, а объекты из разных кластеров различны. Проще говоря, это автоматическая группировка данных без предопределенных правил. Кластеризация и поиск сообществ схожи тем, что не требуют размеченных данных и направлены на выявление внутренней структуры. В графовых данных поиск сообществ можно рассматривать как кластеризацию вершин, а классификация вершин (англ. *node classification*) часто применяется после выделения сообществ для предсказания свойств узлов.

3.3 Классический подход к поиску сообществ через поток Риччи

Обнаружение сообществ — задача выявления тесно связанных групп (сообществ) в сети, где узлы внутри группы взаимодействуют друг с другом чаще, чем с остальной сетью. В статье "Community Detection on Networks with Ricci Flow"([3]) обсуждается, как эту проблему можно решить с помощью алгоритма потока Риччи, отрезая ребра с большим весом после определенного числа итераций. В работе также показаны результаты применения алгоритма на небольших графах. Формально, алгоритм описывается псевдокодом 1.

В статье рассматриваются невзвешенные графы, поэтому все изначальные веса выставляются равными 1. Также стоит отметить критерий остановки, возникающий из предположения, что граф все же сходится к метрике постоянной кривизны.

3.4 Особенности нашего исследования

В нашей работе мы хотим использовать алгоритм из прошлого пункта для кластеризации набора данных, который изначально не имеет структуры графа. Это исследование будет отличаться от работ по обнаружению сообществ по нескольким причинам:

1. Размер данных. В статье наибольший рассматриваемый граф состоит из 1,007 узлов, тогда как в анализе данных мы обычно работаем с десятками тысяч точек данных.

Algorithm 1 Обнаружение сообществ потоком Риччи

Require: Граф $G = (V, E, w)$, скорость обучения η, порог ϵ, максимальное число итераций T

Ensure: Набор сообществ $\{C_1, ..., C_k\}$

Инициализировать веса рёбер $w_{ij}^{(0)} = w_{ij}$ для всех $(i, j) \in E$
for $t = 1$ до T **do**
  for all $(i, j) \in E$ **do**
    Вычислить вероятностные меры μ_i, μ_j для окрестностей вершин
    Рассчитать расстояние Вассерштейна $W_1(\mu_i, \mu_j)$
    Вычислить кривизну $\kappa_{ij}^{(t)} \leftarrow 1 - W_1(\mu_i, \mu_j)/w_{ij}^{(t-1)}$
    Обновить вес: $w_{ij}^{(t)} \leftarrow w_{ij}^{(t-1)} - \eta \cdot \kappa_{ij}^{(t)} \cdot w_{ij}^{(t-1)}$
    Применить ограничение: $w_{ij}^{(t)} \leftarrow \max(w_{ij}^{(t)}, 0)$
  end for
  Вычислить среднее изменение весов $\Delta^{(t)} = \frac{1}{|E|} \sum_{(i,j)\in E} |w_{ij}^{(t)} - w_{ij}^{(t-1)}|$
  if $\Delta^{(t)} < \epsilon$ **then**
    break
  end if
end for
Построить новый граф $G' = (V, E')$ где $E' = \{(i, j) \in E \mid w_{ij}^{(T)} < \epsilon\}$
Найти компоненты связности G': $\{C_1, ..., C_k\}$
return $\{C_1, ..., C_k\}$

2. Вариативность построения графа. В классической проблеме обнаружения сообществ мы работаем с уже известным графом и не имеем другой информации. В нашей задаче сперва нужно определить, как построить граф на основе данных.

3. Наличие правильных меток. В графе вершины обычно однородны, и для оценки правильности разделения на сообщества используются метрики, использующие структуру графа, такие как, например, модулярность. При работе с размеченными датасетами мы сможем оценить производительность алгоритма с помощью оценки правильности классификации после разбиения на сообщества.

4 Настройка эксперимента

4.1 Построение графа

В этом пункте мы обсудим, как мы строим граф по датасету, какие здесь есть степени свободы и обоснуем наш способ.

Коротко построение графа можно описать следующим образом. Мы выбираем некоторое представление данных и вводим на нем метрику, которая позволяет оценить сходство между объектами. Затем мы вычисляем все попарные расстояния в соответствии с выбранной метрикой. После этого мы устанавливаем пороговое значение и строим граф, в котором вершины соответствуют данным, а ребра соединяют только те пары вершин, расстояние между которыми меньше установленного порога.

Первый шаг — получение представления. Поиск преобразования объектов (слов, изображений, аудио, документов и т.д.) в числовые векторы фиксированной длины, которые

сохраняют их смысл, свойства и взаимосвязи. В этой работе мы рассмотрим два варианта представления MNIST и один для ArXiv, подроюнее об этом мы поговорим в описаниях экспериментов.

Второй шаг — выбор метрики. Выбрав представление мы можем использовать любые меры сходства векторов. Мы рассмотрели три основных типа метрик для задач кластеризации и классификации:

1. Метрики L_1, L_2 и L_inf
2. MSE (Среднеквадратичная ошибка)
3. Косинусное сходство

При выборе метрик сходства представлений, в машинном обучении пользуются базовым правилом[6]: использовать ту метрику, которая применялась при обучении вашей модели эмбеддингов. Мы будем следовать этому правилу для представлений построенных с помощью обучаемых моделей и подбирать метрику вручную для не оптимизированных представлений.

Третий шаг — метод фильтрации. Мы строим граф с вершинами из набора данных, добавляя ребра, когда расстояние между ними ниже определенного порога.

Сначала мы попробовали поиск порога полу-контролируемым образом, анализируя результат. Однако для масштабирования результатов мы искали автоматические методы выбора порога, такие как выбор некоторого квантиля всех расстояний или нахождение минимального порога, который делает граф связным (через построение минимального остовного дерева). Тут надо заметить что на несвязном графе поток Риччи будет менять веса на каждой компоненте независимо, поскольку между компонентами отсутствуют ребра, и, следовательно, нет связи для «перетекания» кривизны между ними. Таким образом, поток Риччи на несвязном графе будет работать по компонентам связности раздельно, не влияя друг на друга. По этой причине мы выбрали в качестве порога минимальное ребро в минимальном остовном дереве. Этот выбор обеспечивает связность, не создавая лишних ребер затрудняющих вычисления.

4.2 Поток Риччи

Далее мы использовали алгоритм потока Риччи из библиотеки GraphRicciCurvature. Мы подаем ему граф с весами ребер, равными расстоянию между вершинами. В алгоритме есть несколько параметров:

1. iterations - максимальное число итераций потока, переменная T из псевдокода к классическому подходу. Значение по умолчанию 20.
2. step - параметр η регулирующий скорость изменения кривизны. Значение по умолчанию 1.
3. delta - параметр определяющий критерий остановки потока, переменная ϵ из псевдокода к классическому подходу. Значение по умолчанию 0.001.

Так как в наших экспериментах графы имеют большее число вершин, чем примеры, на которых обычно вычисляют поток Риччи, мы увеличили параметр iterations так, чтобы граф точно сходился к постоянной кривизне. Остальные параметры мы оставили дефолтными.

4.3 Процедура срезки

После потока Риччи веса ребер изменяются. Классический метод обнаружения сообществ предполагает удаление ребер со слишком большим весом — считается, что эти ребра соединяют разные кластеры. Порог отсечения — это еще один параметр, который нам нужно определить.

В изученных работах прямых указаний по выбору порога после потока Риччи нет, однако можно выделить общие подходы и рекомендации из теории кластеризации и анализа графов, которые применимы и в данном контексте.

1. Анализ распределения весов ребер после потока Риччи. После применения потока Риччи ребра графа получают новые веса. Для выбора порога можно проанализировать распределение этих весов: Например можно выделить естественные разрывы или "скачки"в распределении весов ребер. Также можно использовать статистические методы, например, порог на основе среднего значения, или квантилей.

2. Использование методов кластеризации и оптимизации. Можно рассматривать задачу как оптимизацию качества разбиения графа, например, максимизацию модулярности или минимизацию некоторой функции потерь. Порог выбирается так, чтобы после удаления ребер качество разбиения было максимальным. Для этого применяются: Поиск оптимального порога с помощью перебора или эвристических алгоритмов. Методы, основанные на задачах p-медиан или других задачах оптимизации кластеризации, которые учитывают вес ребер и структуру графа.

Мы будем пользоваться первым методом, чтобы сделать дальнейшую оценку качества разбиения более честной. Мы будем выбирать порог в полу-ручном формате, опираясь на распределение весов и обосновывая выбор для каждого эксперимента.

5 Эксперимент с датасетом MNIST

5.1 Датасет

Набор данных MNIST (Modified National Institute of Standards and Technology) — это классический benchmark в области машинного обучения и компьютерного зрения, который десятилетиями служит отправной точкой для разработки и тестирования алгоритмов распознавания изображений. Он состоит из 70 000 черно-белых изображений рукописных цифр от 0 до 9, разбитых на тренировочную (60 000) и тестовую (10 000) выборки. Каждое изображение представляет собой матрицу 28×28 пикселей, где значение каждого пикселя — целое число от 0 до 255, соответствующее градации серого: 0 означает белый цвет, 255 — черный.

Этот датасет был создан на основе более ранней базы NIST, но тщательно обработан: цифры нормализованы по размеру и центрированы, что значительно снижает уровень шума и упрощает задачу классификации. Благодаря этому MNIST долгое время оставался золотым стандартом для проверки эффективности алгоритмов — от простых методов вроде логистической регрессии до сложных архитектур нейронных сетей. Например, современные модели, такие как ResNet или CapsNet, демонстрируют на MNIST точность выше 99%, что близко к человеческому уровню распознавания.

В нашем эксперименте мы рассматриваем не весь датасет MNIST. Вычислительные мощности обычного компьютера не позволяют вычислять поток Риччи на графе такого размера, поэтому для экспериментов мы использовали функцию $get\_indices(data, n)$, которая собирает из датасета $data$ выборку размера n, так, что все таргеты (в данном случае цифры 0-9) представлены в равной степени.

5.2 Эксперимент с исходными данными

Изначально каждая картинка задается матрицей. В нашем первом эксперименте мы уложили строки матрицы последовательно в один вектор и рассматривали этот вектор как представление картинки. Этот подход имеет очевидные недостатки:

1. Потеря локальной структуры изображения.

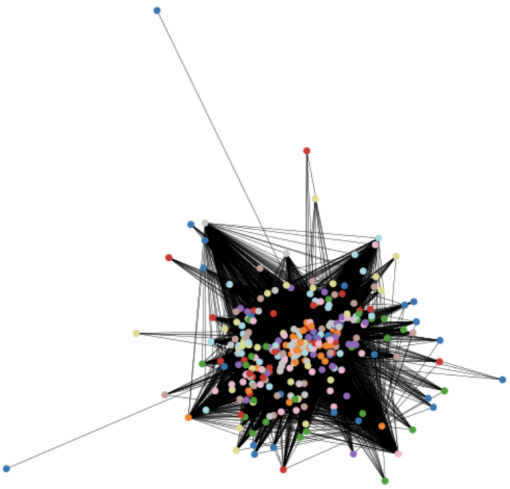
   Изображения имеют важные локальные корреляции (например, соседние пиксели часто связаны). При развертке в вектор эта пространственная информация теряется, и алгоритму (например, классификатору) сложнее выявлять такие закономерности.

2. Неинвариантность к геометрическим преобразованиям.

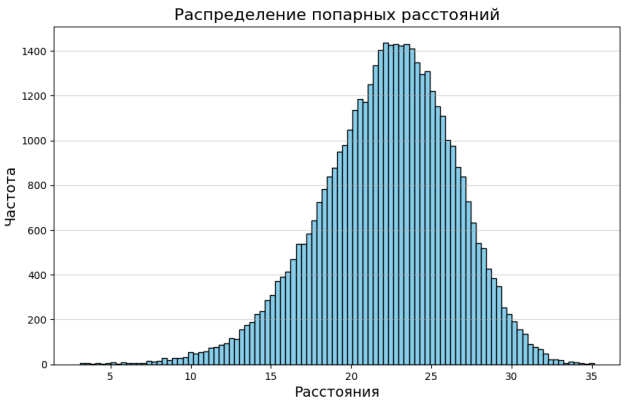
   Сдвиг, поворот, масштабирование: Даже небольшое изменение положения объекта на изображении приведет к совершенно другому вектору, так как пиксели сместятся в другую часть вектора. Отражение, деформации: Аналогичные проблемы возникают при других преобразованиях.

Тем не менее, этот метод позволил нам протестировать и отладить эксперимент. В качестве метрики мы пробовали все перечисленные ранее метрики, за неимением больших отличий остановились на Евклидовом расстоянии.

На визуализации исходного графа (рис. 1a) трудно выделить хорошее разбиение на сообщества. Это подтверждается так же распределением попарных расстояний (рис. 1b). Если распределение попарных расстояний между точками в пространстве близко к нормальному, это может свидетельствовать о том, что координаты этих точек были сгенерированы гауссовским шумом. Рассмотрим этот вопрос подробнее.



(a) Граф до потока Риччи



(b) Распределение попарных расстояний на выборке MNIST размера 300

Рис. 1: Построение графа для исходных данных MNIST

Пусть у нас есть набор точек $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ в d-мерном пространстве, где каждая координата точки $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})$ является независимой случайной величиной с нормальным распределением:

$$x_{ij} \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, N;\ j = 1, \ldots, d$$

Разность координат $(x_{ik} - x_{jk})$ также имеет нормальное распределение:

$$x_{ik} - x_{jk} \sim \mathcal{N}(0, 2\sigma^2)$$

Следовательно, если считать расстояние по Евклидовой метрике, квадрат расстояния r_{ij}^2 представляет собой сумму квадратов независимых нормальных случайных величин:

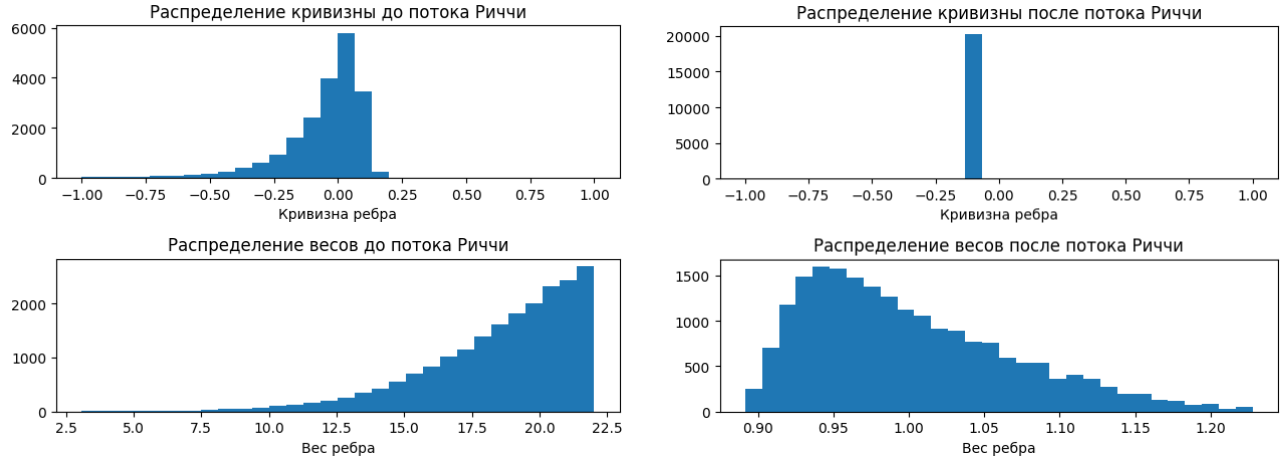
$$r_{ij}^2 = \sum_{k=1}^{d}(x_{ik} - x_{jk})^2 = 2\sigma^2 \sum_{k=1}^{d} z_k^2, \quad z_k \sim \mathcal{N}(0,1)$$

При больших d (благодаря центральной предельной теореме) распределение r_{ij}^2, а значит и r_{ij}, стремится к нормальному. В частности:

$$r_{ij} \approx \mathcal{N}(\sigma\sqrt{2d}, \sigma^2) \tag{3}$$

Таким образом, наблюдаемое нормальное распределение попарных расстояний между точками может служить индикатором того, что их координаты были сгенерированы гауссовским шумом, особенно в пространствах высокой размерности. Конечно, обратное утверждение не всегда верно - существуют и другие процессы, способные порождать нормально распределенные расстояния, однако структура с явным разделением на сообщества, как мы увидим дальше, обычно имеет другой вид.

Изначальное распределение весов в графе это обрезанное распределение с картинки ниже. После потока Риччи (рис. 2) распределение весов похоже на Бета-распределение с параметрами $\alpha = x, \beta = 2x$. При этом кривизна стабилизировалась (здесь нам хватило 50 итераций потока).



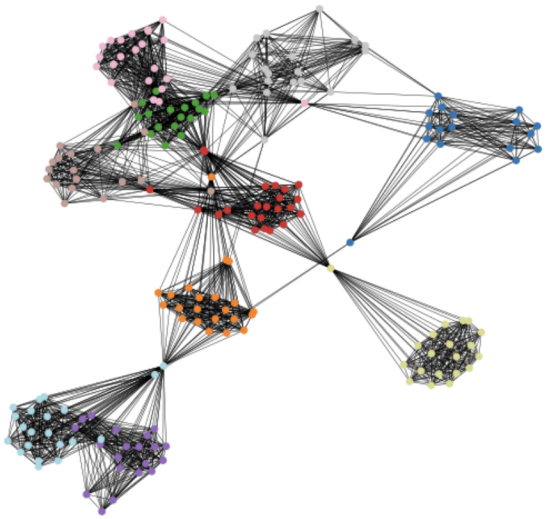
(a) Статистика графа до потока Риччи (b) Статистика графа после потока Риччи

Рис. 2: Сравнение статистики графа до и после потока Риччи

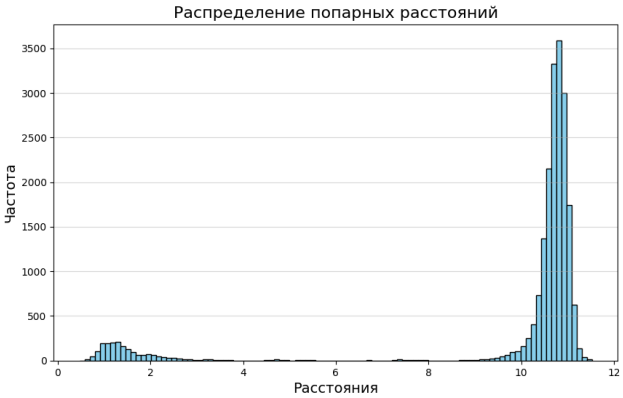
Такое распределение весов после потока не дает нам хорошего порога для срезки ребер, поэтому мы перебрали разные значения потока, примеры результатов есть в разделе Appendix A. Ни один из порогов не дал нам хорошей кластерной структуры, что логично в предположении случайного шума.

5.3 Эксперимент с векторным представлением

Для улучшения структуры данных мы использовали векторные представления MNIST, полученные с помощью Vision Transformer (размерность 768). Vision Transformer (ViT) разбивает изображение на патчи и обрабатывает их через трансформерные слои, что позволяет эффективно учитывать глобальные зависимости между различными частями изображения. В нашем случае ViT учится сближать представления схожих изображений и отдалять разные по L2-норме, поэтому далее мы будем использовать Евклидову метрику.



(a) Граф до потока Риччи

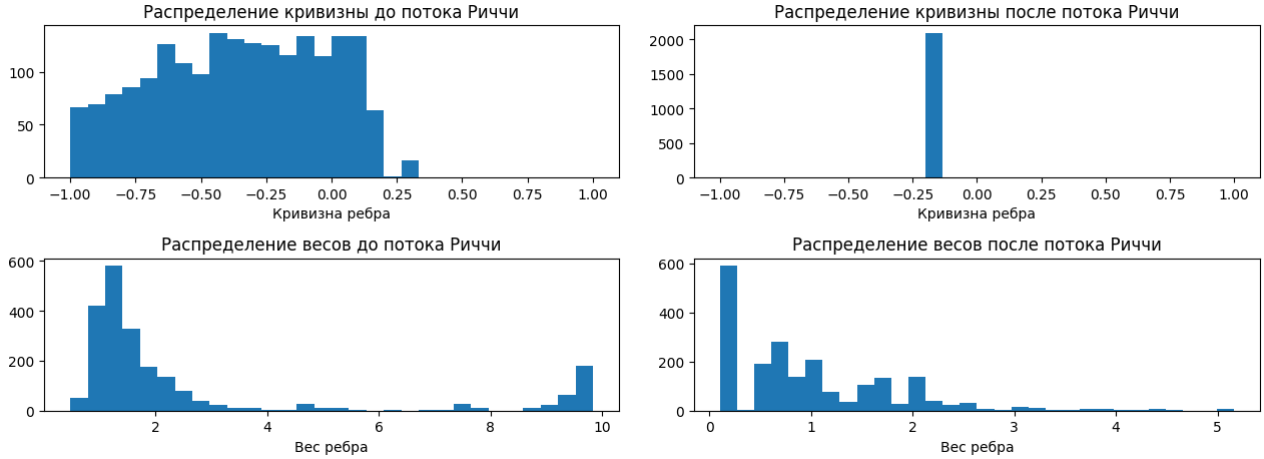


(b) Распределение попарных расстояний на выборке MNIST размера 200

Рис. 3: Построение графа для векторного представления MNIST

Распределение расстояний (рис. 3b) демонстрирует совсем другую структуру по сравнению с исходными данными. Два явных пика делают ребра на два вида: внутри сообщества (левый пик) и между сообществами (правый пик). Ребер первого вида: $\frac{n/10*(n/10-1)}{2} \sim n^2/200$, а второго вида: $\frac{n*9n}{2} \sim 4,5n^2$. Эти наблюдения вместе с визуализацией (рис. 3a) говорят нам о том, что изначальные данные хорошо преобразованы и дают близкие представления внутри одного класса. Более того, для этого представления существует порог, который делит изначальный граф по классам, однако для дальнейших наблюдений нам нужна связность.

Изначально распределение веса в графе имеет так же два пика (рис. 4), однако после применения потока Риччи наблюдаются следующие изменения.



(a) До потока Риччи

(b) После потока Риччи

Рис. 4: Статистика графа для векторных представлений MNIST

Кривизна становится постоянной, а веса уменьшаются теряя изначальное явное разделение. После потока перебор разных порогов срезки уже не дает разбиения по классам, что показано в разделе Appendix B.

5.4 Выводы

В этом исследовании мы не планировали выявления принципиально новых наблюдений о датасете MNIST, который в данный момент уже очень подробно исследован, однако мы можем выделить некоторые наблюдения о потоке Риччи, которые могут быть полезны для повторения алгоритма на других данных, и подытожить результаты экспериментов.

Ключевые результаты и наблюдения:

1. Исходные данные MNIST демонстрируют распределение попарных расстояний, близкое к нормальному, что может свидетельствовать об отсутствии явной кластерной структуры. Поток Риччи не смог выделить значимые сообщества, что согласуется с гипотезой о шумовом характере данных в исходном пространстве.

2. Векторные представления ViT показали более выраженную структуру: распределение расстояний имело два пика, соответствующих внутриклассовым и межклассовым связям. Однако после применения потока Риччи веса рёбер изменились таким образом, что разделение на сообщества стало менее явным, и подбор порога срезки не позволил восстановить исходные классы.

3. Поток Риччи чувствителен к начальной структуре графа и качеству представления данных. Для успешного выделения сообществ необходимо, чтобы исходные расстояния между объектами уже содержали некоторую структурную информацию.

4. В случаях, когда данные изначально не имеют графовой структуры, критически важен этап построения графа: выбор метрики, порога связности и метода фильтрации рёбер.

5. Наблюдаемое "размытие"кластеров после потока Риччи в векторных представлениях ViT может быть связано с тем, что алгоритм стремится сделать кривизну постоянной, что не всегда соответствует задаче кластеризации.

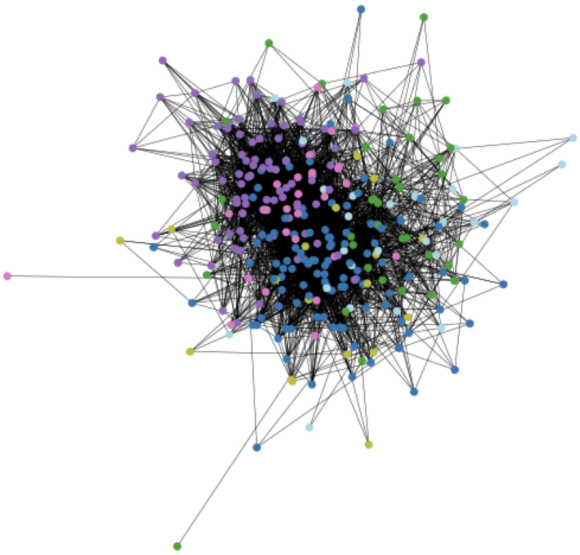
6 Эксперимент с данными arXiv

6.1 Датасет

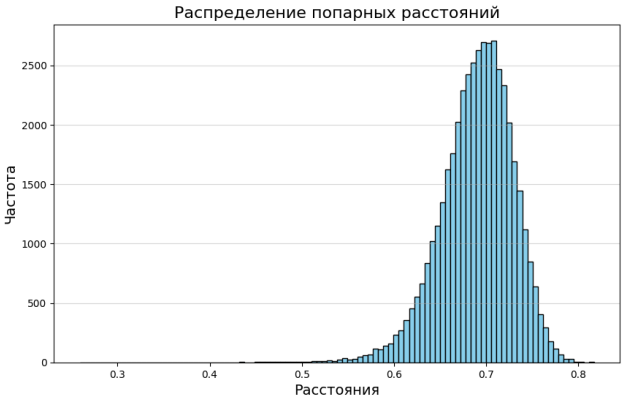
Для проверки нашего подхода на сложных данных мы использовали набор научных статей из arXiv в области компьютерных наук (Computer Science), предварительно обработанных в несколько этапов. Чтобы обеспечить четкую разметку, мы отфильтровали статьи, оставив только те, которые принадлежат к одной из шести ключевых категорий: компьютерное зрение (cs.CV), машинное обучение (cs.LG), обработка естественного языка (cs.CL), искусственный интеллект (cs.AI), нейронные сети (cs.NE) и робототехника (cs.RO). Векторные представления размерности 1536 были получены с помощью модели text-embedding-ada-002 от OpenAI, которая эффективно кодирует семантическое содержание текстов. Этот датасет представляет особый интерес, поскольку, в отличие от MNIST, он обладает более сложной структурой: тематические категории могут пересекаться, а сами статьи часто содержат смешанную тематику. Кроме того, высокая размерность представлений и их семантическая природа позволяют проверить, насколько хорошо метод выделяет сообщества в данных с неочевидными геометрическими свойствами. Исходный набор содержит более чем 250,000 текстов, однако для эксперимента мы выбираем 300 случайных элементов так, что они представлены в выборке пропорционально их представленности в исходном датасете.

6.2 Эксперимент

Распределение расстояний (рис. 5b), как и в случае первого эксперимента с MNIST имеет один пик. Визуализация графа (рис. 5a) тоже не позволяет сделать каких-то наблюдений о кластерной структуре в исходном графе.



(a) Граф до потока Риччи

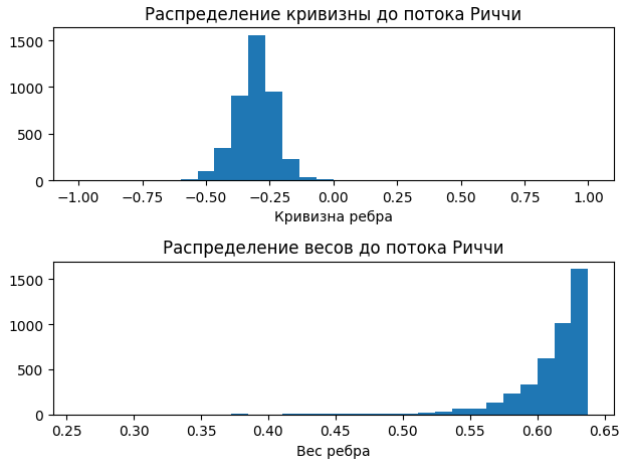


(b) Распределение попарных расстояний на выборке ArXiv размера 300

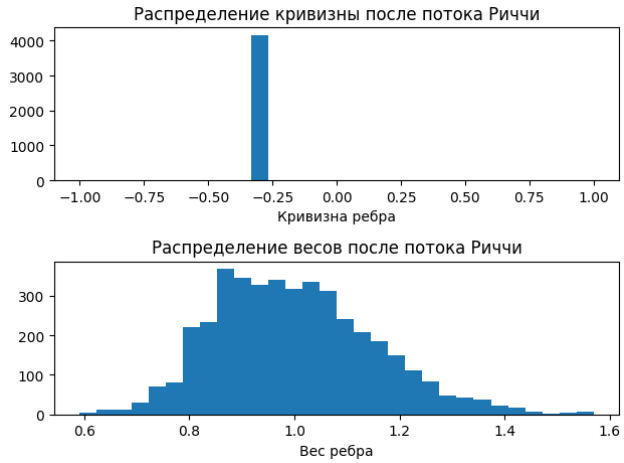
Рис. 5: Построение графа для векторного представления ArXiv

Проанализировав далее статистики веса и кривизны (рис. 6) можно заметить, что гистограмма веса превратилась из распределения с явным пиком в распределение, которое можно интерпретировать как смесь двух распределений с разными средними. В этом предположении мы можем предполагать, что подходящий порог должен лежать в отрезке $(0.8, 1.2)$.

На визуализации графа после срезки с разными порогами (рис. 7) представлены два графа. После срезки с порогом 1.0 граф остаётся очень плотным. Статьи разных тематик сильно взаимосвязаны, хотя мы можем наблюдать новую структуру. Статьи с одинаковыми темами локализуются, почти все цвета формируют достаточно явные сгущения.



(a) До потока Риччи



(b) После потока Риччи

Рис. 6: Статистика графа для данных arXiv

Можно заметить, что гистограмма веса (рис. 6) превратилась из распределения с явным пиком в распределение, которое можно интерпретировать как смесь двух распределений с разными средними. В этом предположении мы можем предполагать, что подходящий порог должен лежать в отрезке $(0.8, 1.2)$.

На визуализации графа после срезки с разными порогами (рис. 7) представлены два графа. После срезки с порогом 1.0 граф остаётся очень плотным. Статьи разных тематик сильно взаимосвязаны, хотя мы можем наблюдать новую структуру. Статьи с одинаковыми темами локализуются, почти все цвета формируют достаточно явные сгущения.

После срезки с порогом 1.2 граф становится существенно разреженным. Появляются более отчётливые кластеры, каждый из которых соответствует определённой тематике. Особенно явно отделяются два крупнейших класса: cs.AI (тёмно-синий) и cs.CV (фиолетовый). Они формируют плотные, хорошо локализованные сообщества. Также наблюдается большое количество "висячих"узлов и небольших компонент, что указывает на удаление слабосвязанных статей, возможно выбросов.

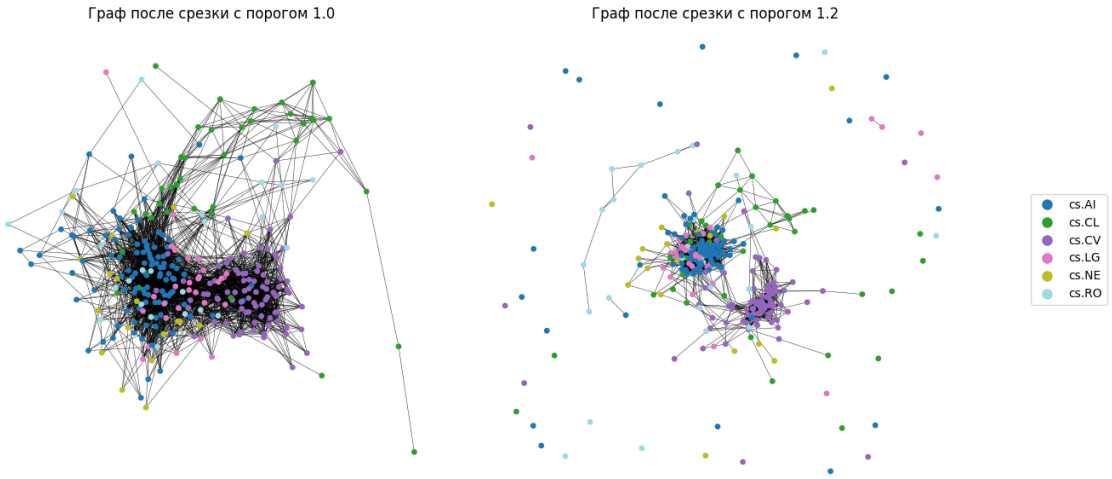


Рис. 7: Графы arXiv после удаления ребер с разными порогами

6.3 Тренд модулярности

Для проверки наших наблюдений воспользуемся численной мерой качества разделения графа на сообщества. Для этого мы построим Modularity Trend for Girvan-Newman Community Detection.

Алгоритм Гирвана-Ньюмана позволяет выявлять сообщества в графе, последовательно удаляя наиболее "важные"рёбра. На каждом шаге алгоритм вычисляет промежуточность (betweenness centrality) для всех рёбер — метрику, которая показывает, через сколько кратчайших путей между всеми парами вершин проходит данное ребро. Ребро с максимальным значением промежуточности удаляется, что приводит к постепенному разрыву связей между естественными сообществами в графе. Этот процесс повторяется итеративно, пока граф не распадётся на отдельные вершины (либо пока не преодолеет максимальное число итераций).

Далее будем считать модулярность, согласно формуле (2). Чем выше модулярность, тем лучше качество разбиения. В коде модулярность рассчитывается для каждого промежуточного состояния графа после удаления очередного ребра.

Визуализация тренда модулярности позволяет определить оптимальное количество сообществ. Как правило модулярность растёт, пока удаление рёбер между сообществами

усиливает кластерную структуру, затем достигает максимума и начинает снижаться, поскольку алгоритм продолжает разрушать уже осмысленные сообщества. Оптимальным считается разбиение с максимальной модулярностью, что хорошо видно на построенном столбчатом графике.

Вычисления (1) показывают, что граф после срезки с порогом 1 лучше делится на сообщества методом Гирвана-Ньюмана.

Таблица 1: Сравнение модулярности графов

Граф	**Число сообществ с лучшей модулярностью**	**Лучшая модулярность**	**Средняя модулярность (1–100 сообществ)**
Исходный граф	21	0.307	0.197
Граф после срезки	10	0.417	0.277

Таким образом, поток Риччи вместе со срезкой улучшил кластеризуемость графа алгоритмом Гирвана-Ньюмана, в чем так же можно убедиться изучив гистограммы в разделе Appendix C.

7 Заключение

7.1 Итоги

В ходе исследования мы изучили применение потока Риччи для кластеризации данных, изначально не имеющих графовой структуры. Основные эксперименты проводились на двух датасетах: MNIST (изображения рукописных цифр) и arXiv (научные статьи по компьютерным наукам). Ниже коротко резюмируем результат.

Поток Риччи эффективен тогда, когда исходные данные содержат скрытую кластерную структуру. В случае MNIST (исходные пиксельные данные) этого не наблюдалось, в то время как для векторных представлений ViT и arXiv структура была более выражена.

Метод показал себя перспективным для семантически сложных данных (arXiv), где классические методы кластеризации могут работать хуже. Однако для простых структур (предобработанный MNIST) он не даёт преимуществ.

Тем не менее, метод требует тонкой настройки и правильного подбора гиперпарметров. Наш алгоритм дает пользователю много свободы, о правильном использовании которой мы говорили в работе.

Код нашего проекта доступен на GitHub.

7.2 Предстоящая работа

Мы планируем продолжать изучать потенциал потока Риччи в анализе данных. Наша дальнейшая исследовательская работа будет касаться двух основных аспектов.

1. Применение потока Риччи для анализа определенных датасетов. Мы планируем использовать наш алгоритм для других датасетов связанных с лингвистикой, биологией и другими науками, получая новые результаты для этих данных. Понимая специфику потока Риччи, мы надеемся получить полезные представления, дальнейшее изучение которых продвинет исследования в других областях.

   В этом направлении мы уже получаем первые результаты. Применив поток Риччи на наблюдениях за мозгом мышей во время движения наши коллеги смогли получить

новую кластерную структуру и сейчас работают над интерпретацией результатов. Это исследование говорит нам о высоком прикладном потенциале потока Риччи.

2. Применение потока Риччи для формализации законов масштабирования нейросетей. Возвращаясь к изначальной идее, мы хотим понять, какие еще структуры мы можем обнаружить изучая потоки Риччи. Используя этот метод мы рассчитываем получать новую информацию, которая позволит упростить и улучшить обучение нейросетей. Мы рассматриваем использование потоков не только как инструмент для решения конкретных задач, но и как перспективное направление в теории глубинного обучения.

Литература

[1] Канторович Л.В., О перемещении масс (2006)

[2] Попеленский Ф.Ю., Комбинаторные потоки Риччи и метрики на триангулированных поверхностях (2024)

[3] Chien-Chun Ni, Yu-Yao Lin, Feng Luo, Jie Gao, Community Detection on Networks with Ricci Flow (2019)

[4] Daniela Leite, Diego Baptista, Abdullahi A. Ibrahim, Enrico Facca, Caterina De Bacco, Community detection in networks by dynamical optimal transport formulation (2022)

[5] Hamilton R.S., Three-manifolds with positive Ricci curvature (1982)

[6] Roie Schwaber-Cohen, Vector Similarity Explained (2023)

[7] Shuliang Bai, Yong Lin, Linyuan Lu, Zhiyu Wang, Shing-Tung Yau, Ollivier Ricci-flow on weighted graphs (2024)

[8] Y. Ollivier, Ricci curvature of markov chains on metric spaces (2009)

A Разные пороги срезки для эксперимента с исходными данными MNIST

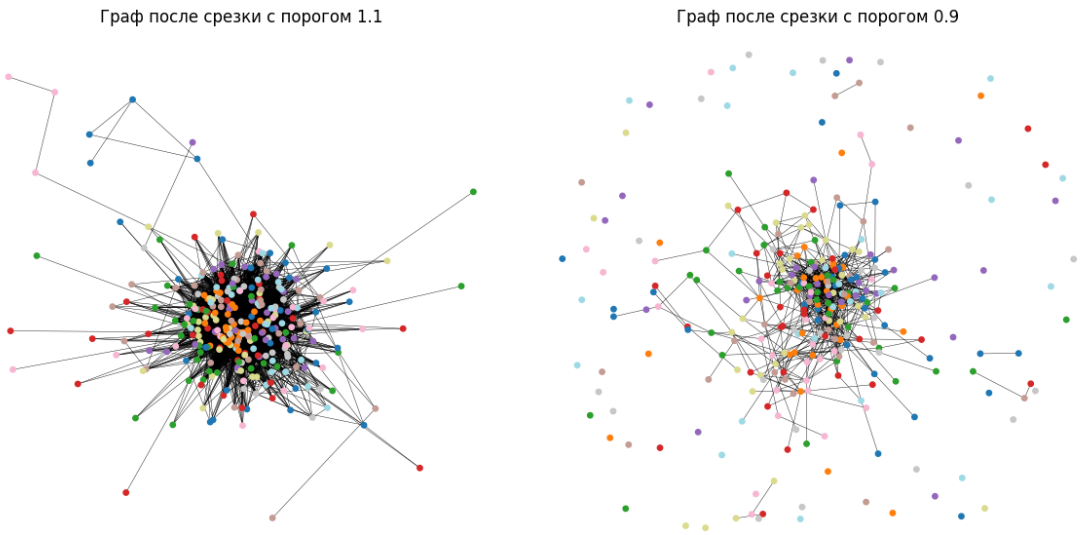


Рис. 8: Графы после удаления ребер с разными порогами

B Разные пороги срезки для эксперимента с векторным представлением MNIST

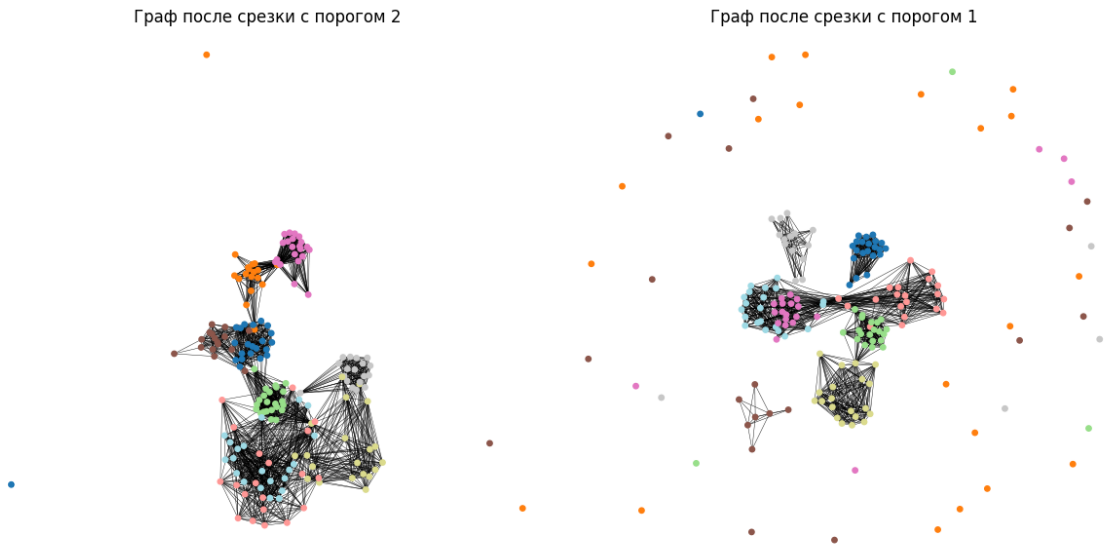


Рис. 9: Графы после удаления ребер с разными порогами

C Графики тренда модулярности для эксперимента с векторным представлением ArXiv

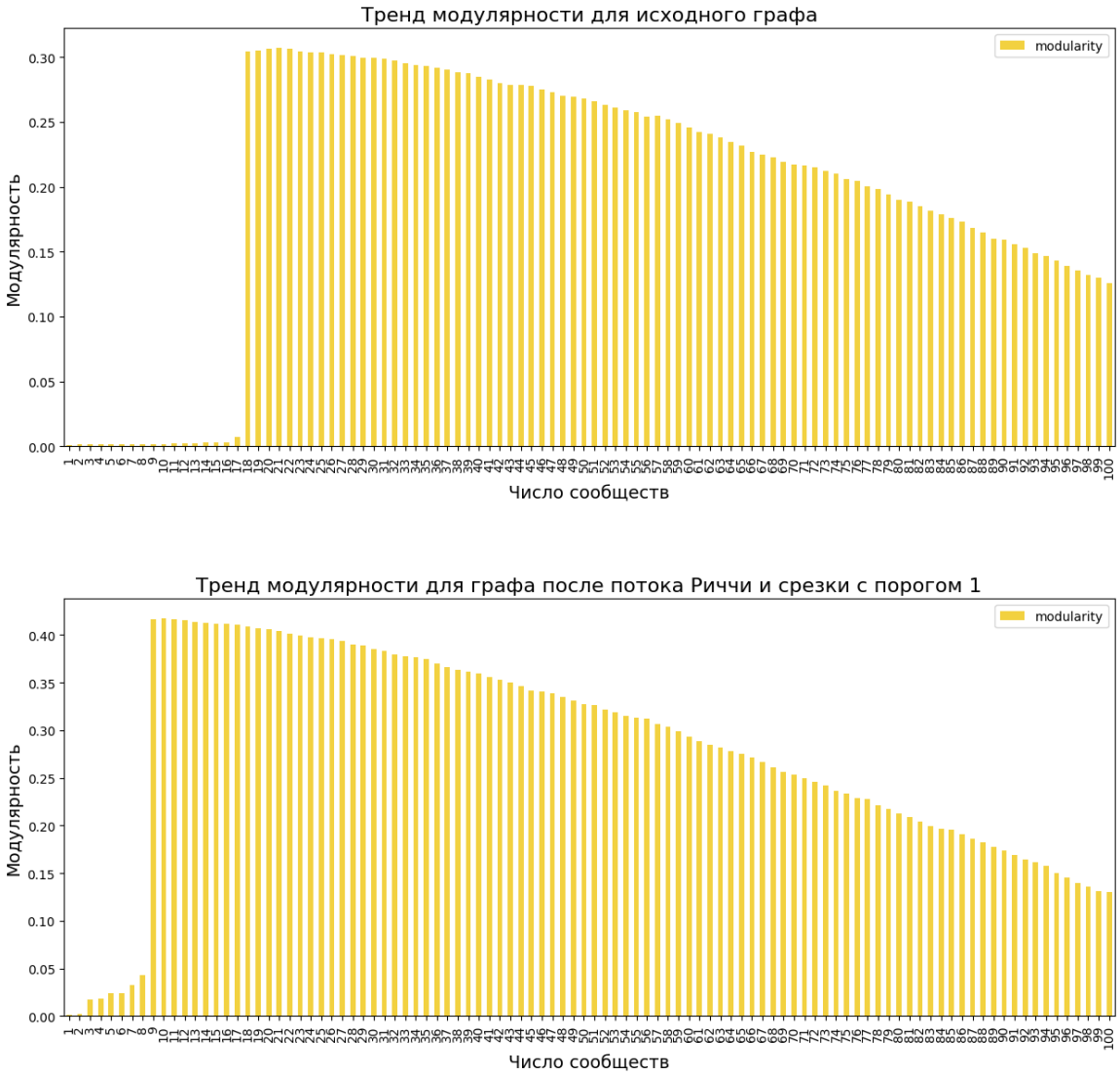


Рис. 10: Тренд модулярности для поиска сообществ алгоритмом Гирвана-Ньюмана